National Park Service
U.S. Department of the Interior

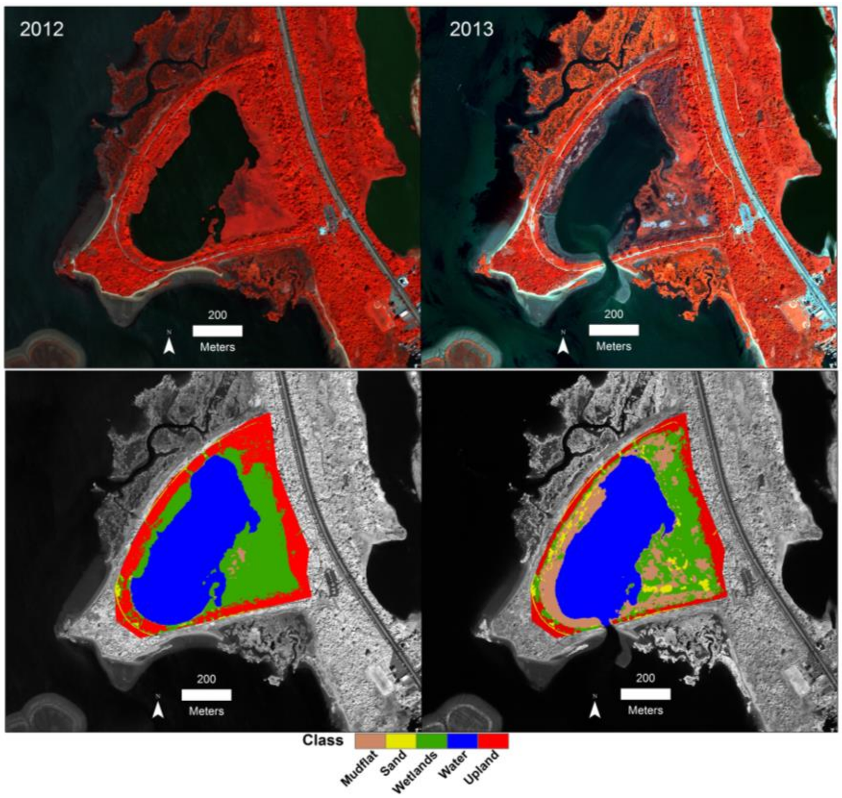
Natural Resource Stewardship and Science

# Object-based Image Analysis (OBIA) of Salt Marshes: Standard Operating Procedures

*Version 1.0*

For

***Monitoring Salt Marshes Using High Spatial Resolution Satellite Imagery for Mapping and Change Detection: Protocol Development for Northeast Coastal Parks***




**August 2018**

**Please cite this document as:** Wang, Y. Q., and A. Campbell. 2018. Object-based image analysis of salt marshes: Standard operating procedures (Version 1.0). Northeast Coastal and Barrier Network, National Park Service, Kingston, RI.

# Contents

# Figures

Page

# Tables

Page

# 1. Required Programs and Packages

The process utilized a variety of image analysis programs, and a combination of R and python scripts (Table 1). The protocol is recommended for a user with familiarity with the GIS. An understanding of remote sensing concepts would be beneficial and allow for greater control over the process. The program utilized in each stage of the protocol is identified in that subsection heading. Many of the required programs can be downloaded with the OSgeo4w a distribution of a number programs, languages and more used in this workflow including QGIS, SAGA GIS, Orfeo Toolbox, GDAL, and Python.

**Table 1.** The required programs and version. All R and python packages used in the process are listed.

| Software | Version | Packages |
|---|---|---|
| Orfeo Toolbox (Inglada & Christophe, 2009) | 5.0.0 | - |
| R (RC Team, 2017) | 3.4.3 | Rgdal |
| | | Caret |
| | | RandomForest |
| | | Spdep<br>Rgeos |
| Python (Python Software Foundation, https://www.python.org/) | 2.7.14 | tkFileDialog |
| | | Json<br>GDAL |
| | | Requests |
| | | Fiona |
| | | Geojson |
| | | Numpy |
| System for Automated Geoscientific Analyses (Saga) GIS (SAGA GIS, http://www.saga-gis.org/ | 2.1.2 | - |
| ERDAS Imagine 2016 (Hexagon Geospatial, 2016) | 16 | - |
| ArcGIS | 10.4.1 | - |

## 2. Preprocessing

Preprocessing is an important step in satellite data analysis. The data is corrected for radiometric and geometric discrepancies. This workflow uses both Orfeo Toolbox and ERDAS Imagine to accomplish these preprocessing steps. Radiometric correction normalizes across dates changing units from Digital Numbers, unique to the date of collection, to reflectance. Geometric correction corrects data locations ensuring all data is correctly aligned. These steps include: (1) radiometric correction which sensor ephemeris to convert DN to top of atmosphere reflectance; (2) geometric rectification to spatially align all the imagery; (3) pansharpening is the process of combining the spectral information of the multispectral bands with the increased spatial resolution of the panchromatic bands; (4) mosaicking, multiple scenes of imagery were collected the mosaicking process allows these to be combined into a single image for further analysis; (5) Masking out upland areas was an important process for the FIIS and Jamaica Bay classifications.

### 2.1 Folder Contents

Worldview-2 imagery is delivered in a folder contents include a license (.txt), the image (.tif), metadata (.xml), a tile file (.til), a jpeg preview (.jpg), and a readme (.txt) (Figure ). The naming scheme is within the readme file and is composed of acquisition time, product info, product id and the file extension. For example the filename in Figure 1 is 12SEP15162527-M2AS-052672331020_01_P001. The image was acquired at September 15, 2012 at 4:25:27 UTC and the M- refers to multispectral and the 2AS is their Level 2 product. The final number is a product ID.

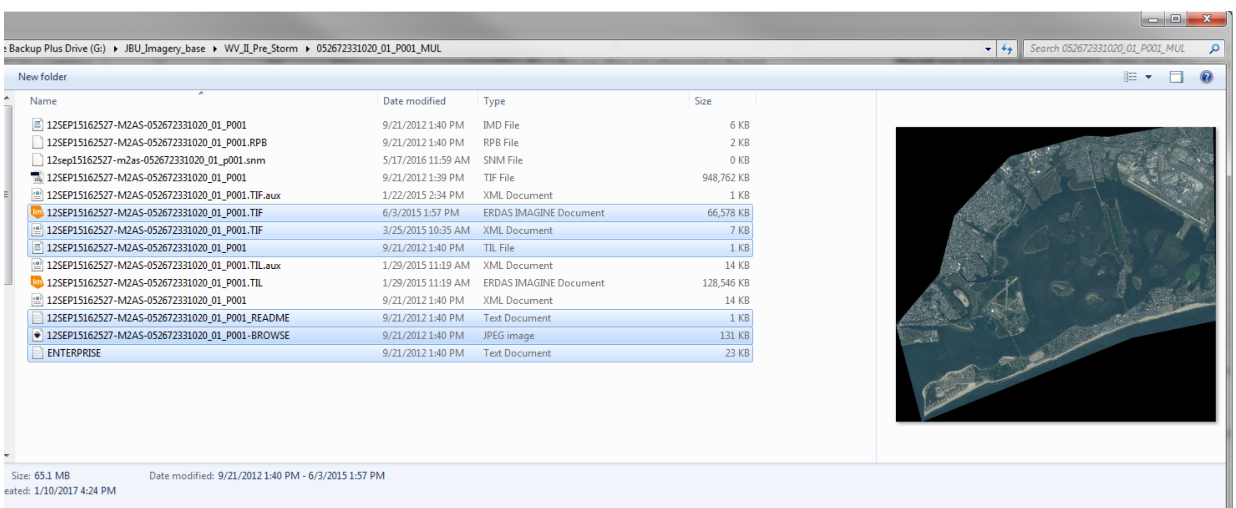



**Figure 1.** The contents of a standard Worldview-2 tile. Base files are highlighted in blue additional files are generated by ERDAS imagine and other image analysis software.

### 2.2 Radiometric Correction (Orefeo Toolbox)

Radiometric correction of the WV-2 data to top of atmosphere reflectance can be done with Orfeo Toolbox. Radiometric correction controls for differences between scenes and normalizes data collected across multiple dates with unique sun angles, sun azimuth, sun elevation and off-nadir angle. Radiometric correction can be conducted with many software packages. Correction equations are utilized from Updike & Comp (2010). Radiometric correction can be run from the OSGeo4W Shell with either the GUI command or the command line approach. The command line approach may have some performance gains, and demonstrated here (Figure 2).

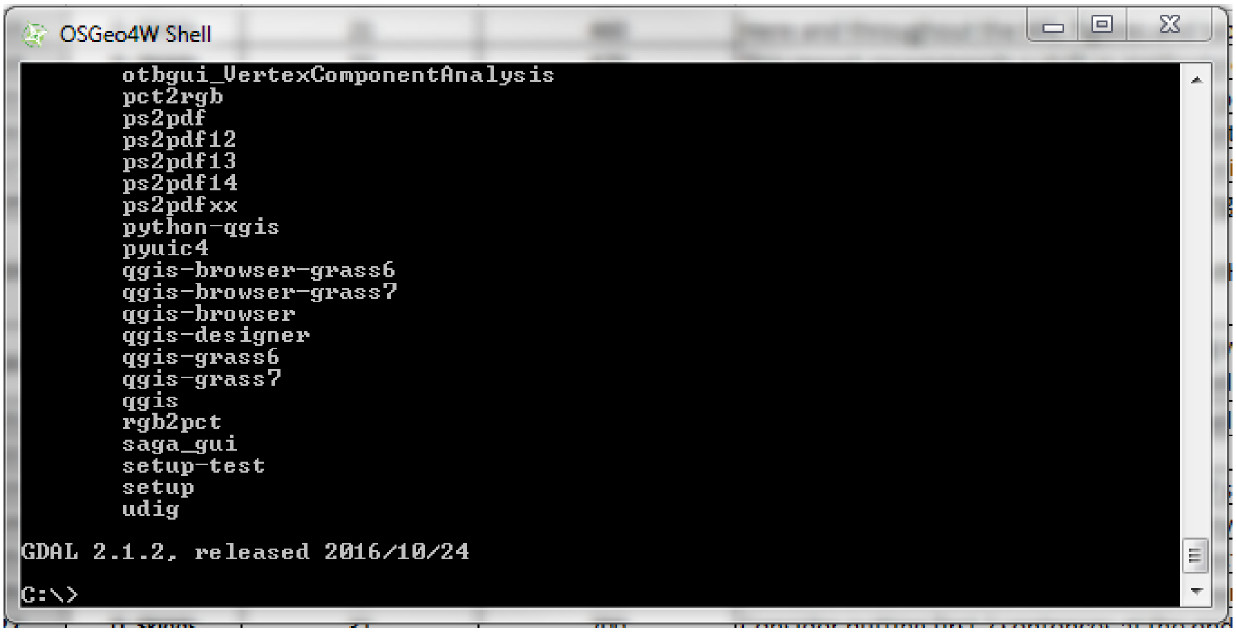



**Figure 2.** The OSGeo4W cmd prompt allowing access to Orfeo Toolbox tools.

**Step 1:**

Command line example:
Step 1: Change directory to the location of interest (the /d is only required when changing drives)

cd /d D:\World_View_2_post\052851566020_01_P001_MUL

**Step 2:** Run the command

otbcli_OpticalCalibration –in 13SEP19162635-M2AS_R2C1-052851566020_01_P001.TIF –level toa –out 13sepM2AS_R2C1_cor.img

First the tool is called, then the input image. The level is set next, which refers the level of radiometric correction this can be either top of atmosphere reflectance or top of canopy, surface reflectance. Finally the output image name.

**Step 3:** Finally it is necessary to run the other images, the tool reads time, and positional data from the sensor metadata which are used to correct the data.

**Note:** Typing otbgui_OpticalCalibration will open a GUI.

## 2.3 Geometric Correction (ERDAS Imagine)

WV-2 satellite imagery has increased the base accuracy of the data products compared to QB-2 satellite. Most WV-2 data products have a geolocation accuracy of within 5m where as QB-2 had 23 m (Digital Globe, 2014). This increase reduces the necessity of geometric correction, still it is a best practice. Geometric correction allows for multiple dates of imagery to be compared and is particularly important for long-term monitoring and change analysis. Geometric correction is best accomplished with ERDAS imagine. The geometric rectification approach remained the same a previous mapping efforts in FIIS and Jamaica Bay (Wang & Traber, 2013; Wang & Christiano, 2005).

Mosaicking the panchromatic and multispectral scenes for each data before geometrically correcting it will save time without a loss of quality, do not mosaic multiple dates before geometrically correcting.

## 2.4 Mosaicking (ERDAS Imagine)

The first mosaic only includes those images collected from a single swath, or pass of the sensor over an area (Figure 3). Jamaica Bay was collected in two swaths in 2013 and a single swath in 2012. Worldview-2 swath is 16.4 km at nadir. FIIS required 4 swaths and ASIS required a single swath to cover the island due to the barrier islands north to south orientation. Each swath of multispectral and panchromatic are mosaicked together first.

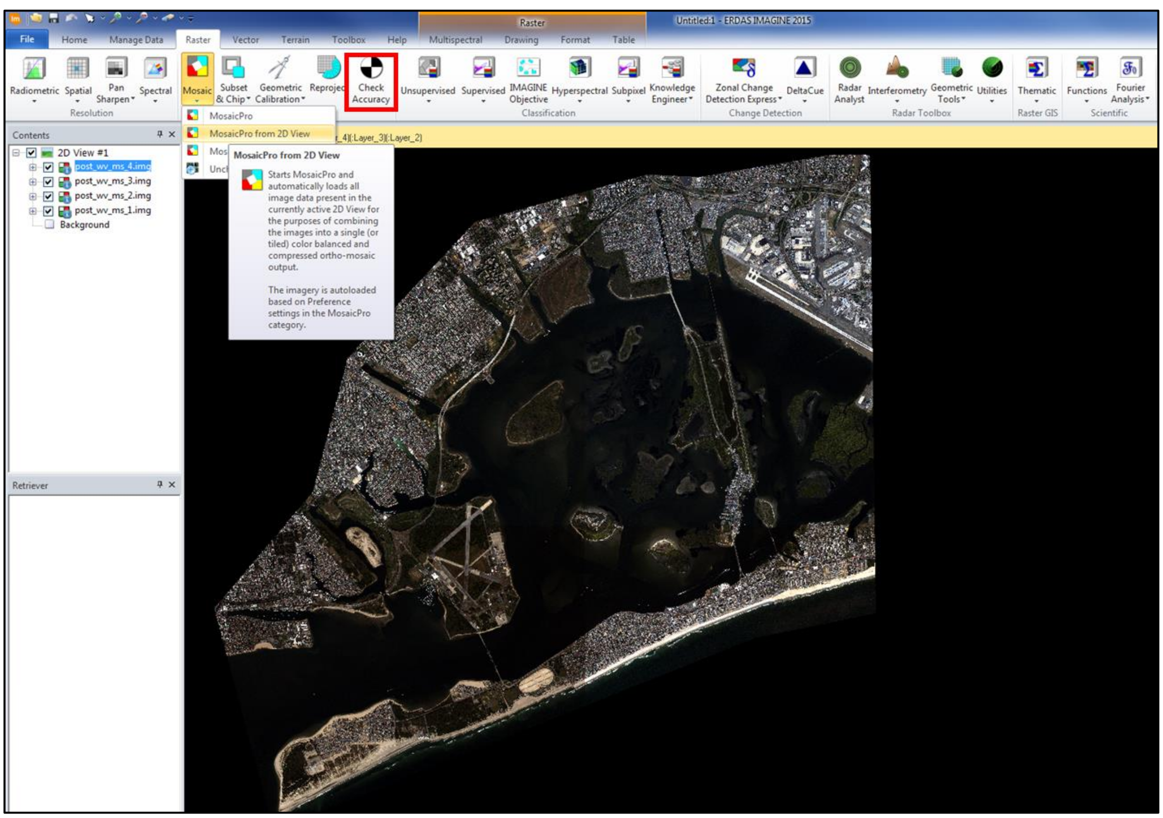



**Figure 3.** Shows multispectral WV-2 data for Jamaica Bay, NY. Ready for mosaicking.

**Step 1:** Open each image to be mosaicked into a 2d view window, the standard visualization window of ERDAS imagine 2015. Images can be opened by right clicking then selecting open raster layer or selecting file then open.

**Step 2:** Select the MosaicPro from 2d View. This is found in Raster -> Mosaic -> MosaicPro from 2d View. This approach directly imports images from 2d viewer to the Mosaic application.

**Step 3:** MosaicPro will open with all images from the 2dview imported. Next the order of the images is very important, these are all from a single swath of data and as such fit perfect without seamlines. In multi-scene mosaics, the correct order of the images is important to minimize overlap. The methods for changing order are shown in red. In blue is the color correction dialog prompt, since we corrected to reflectance no color correction should be conducted. The fx button to the right of color correction determines seamline functions. The red lightning bolt processes the mosaic (Figure 4).

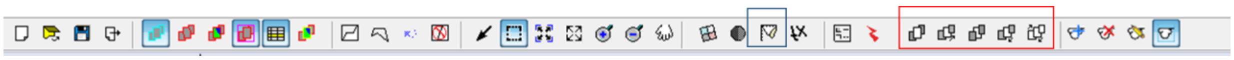


**Figure 4.** The figure shows the mosaic toolbar in Erdas Imagines MosaicPro extension.

**Step 4:** Follow the same approach and mosaic the panchromatic images of the same extent.

## 2.5 Pansharpening (ERDAS Imagine)

The hyperspherical color space merge (HCS), the recommended pansharpening technique by Digital Globe for the WV-2 sensor system was utilized (2010). HCS was also found to be capable of pansharpening all bands of the WV-2 multispectral imagery.

**Step 1:** Raster -> Pan Sharpen -> HCS Resolution Merge (Figure 5).

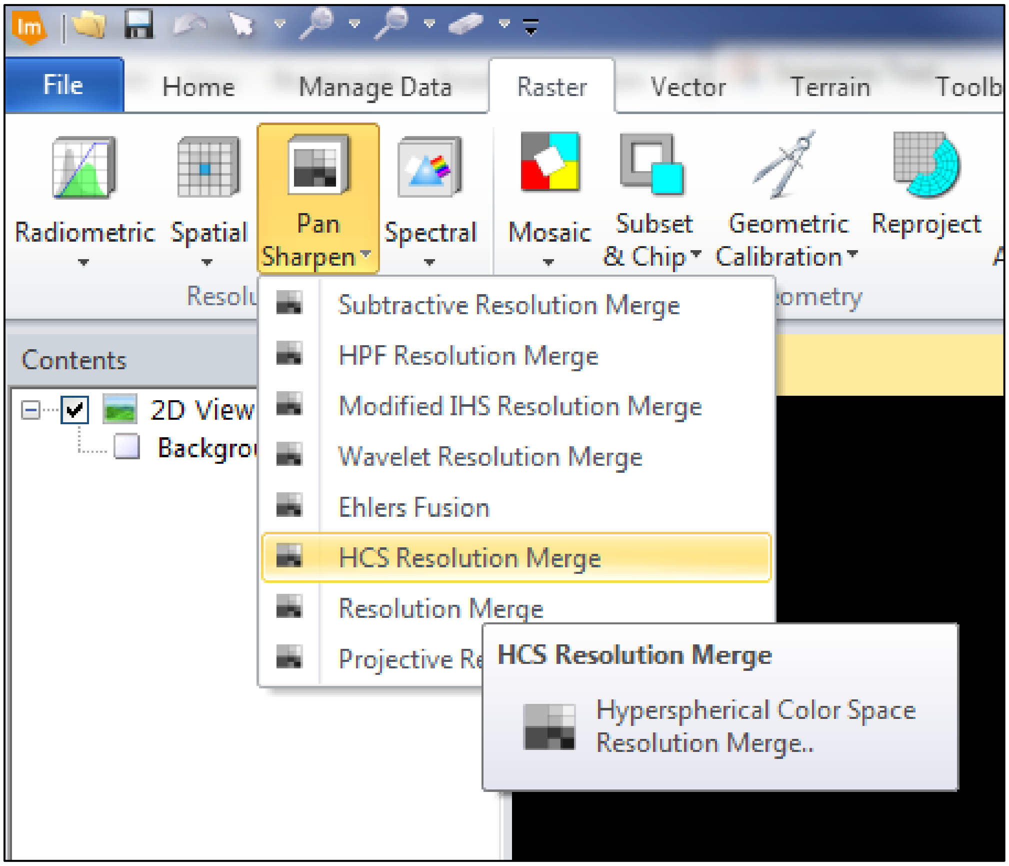



**Figure 5.** The HCS Pan Sharpening technique.

**Step 2:** Select pan chromatic input 1, and multispectral as input 2. Smoothing filter 1, resampling nearest neighbor. The defaults are appropriate (Figure 6).

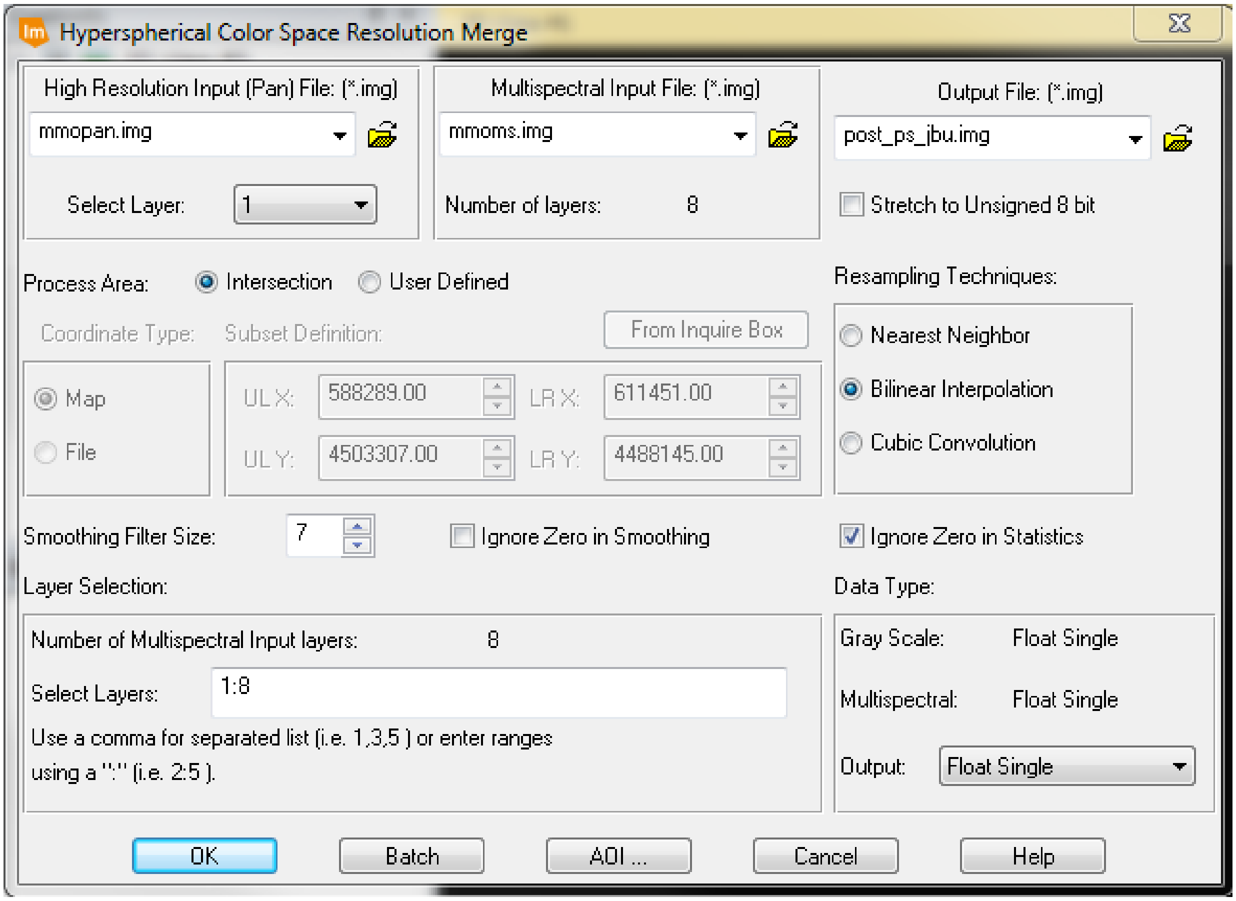



**Figure 6.** The HCS pan sharpening window, with appropriate values.

## 2.6 Mask (ERDAS Imagine)

Next a mask is created to remove areas beyond a studies interest such as urban areas and extensive areas of water. A mask can be created within ERDAS Imagine.

**Step 1:** Right click within a 2d view window, opening a menu (Figure 7).

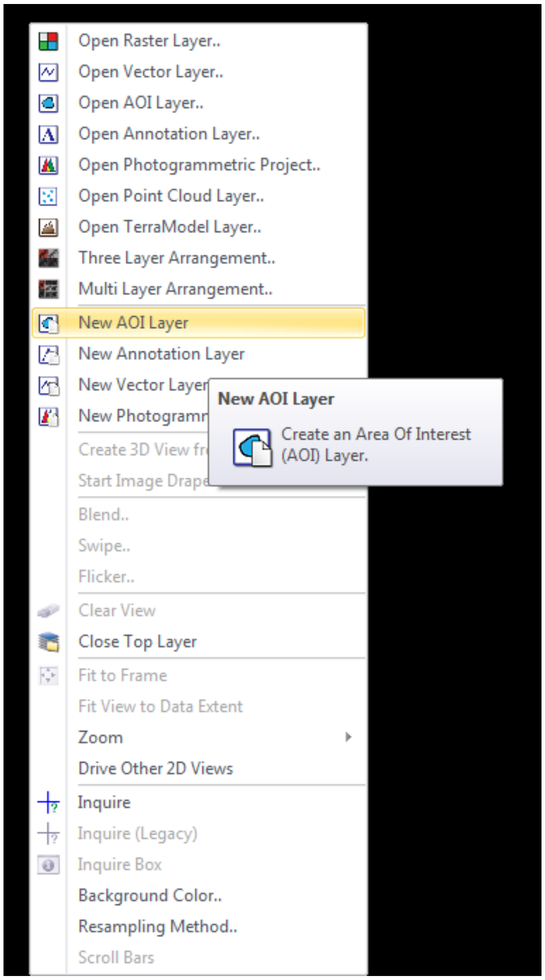



**Figure 7.** The 2d view menu which can be used to create a new AOI layer.

**Step 2:** Select New AOI Layer (Imagery data of the area must be in the 2d view)

**Step 3:** The new AOI will be empty and in the 2d view table of contents (Figure 8).

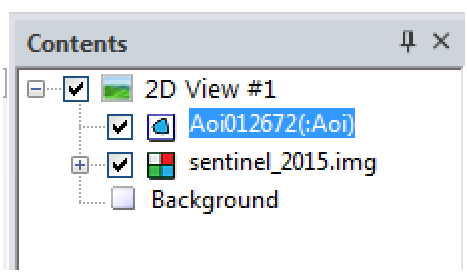



**Figure 8.** Table of contents including imagery data and a new AOI.

**Step 4:** Individual AOI can then be created with the drawing tools in ERDAS Imagine. This is accomplished by first selecting the layer from the Contents window then selecting the Drawing tab -> and the click the  within the Insert Geometry section (Figure 9).

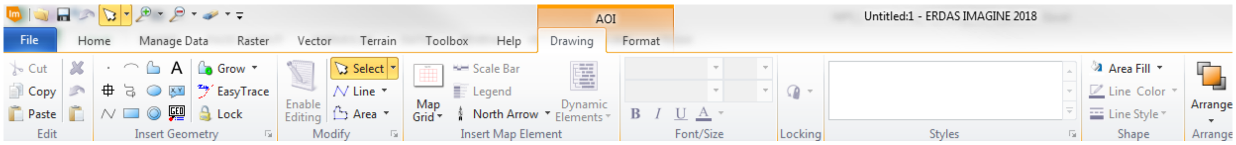



**Figure 9.** The menu for AOI creation in ERDAS Imagine.

**Step 5:** Once you have selected  left click to create vertices and double click to finish the AOI polygon (Figure 10).

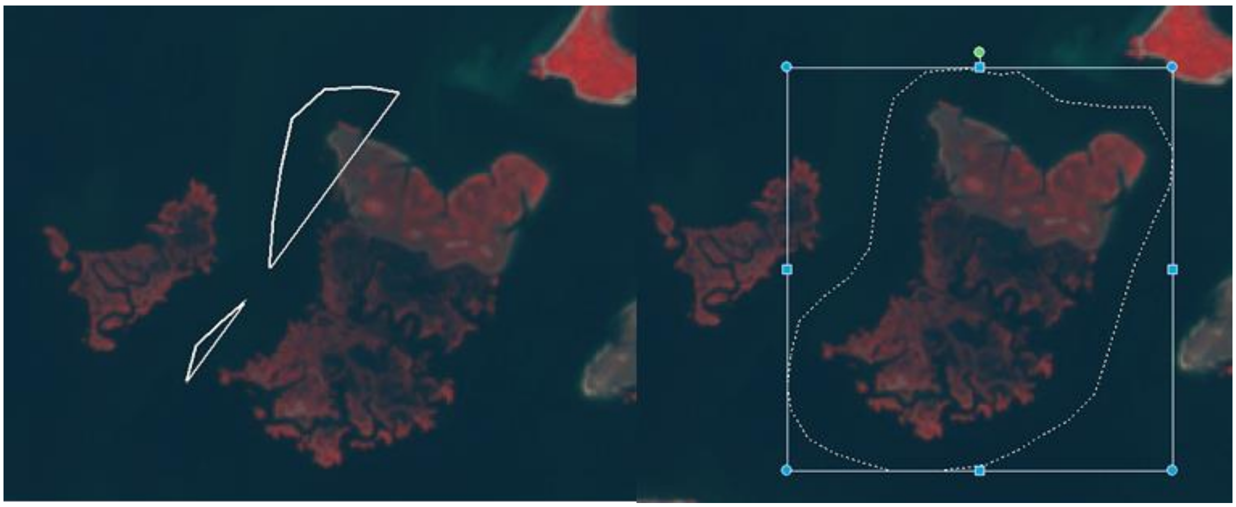


**Figure 10.** The AOI creation process panel 1 shows the start of the process and panel 2 the finished AOI.

**Step 6:** The final AOI will likely be composed of several polygons (Figure 11). If you intend to classify the entire island similar to the ASIS example, still use an AOI to remove excessive areas of water.

**Figure 11.** The completed AOI file for Jamaica Bay.

# 3. Segmentation

The script steps are broken down as follows and complete script can be found in the code folder accompanying this document.

1. Segmentations for each of the minimum object size and spectral radius parameters are created.
2. The resulting segmentation raster's are converted to polygons with GDAL
3. The mean and variance of each object is calculated for the NIR1 band of the WV-2 sensor
4. The data is read into R
5. Variance is normalized by the area for each image object
6. Moran I is calculated for each segmentation with the spdep package in R
7. The area normalized variance and Moran I are combined for each of the segmentations
8. The segmentation with the minimum value for these two parameters is considered to have the lowest intrasegment homogeneity and intersegment heterogeneity i.e. be most appropriate for the landscape.
9. The script then calculates local segmentation quality for each of the best segmentations objects.
    a. Calculated with variance
    b. local Moran I
10. The bottom 20% and top 20% of this combined measure are considered over and under segment respectively
11. These segments are then exported and used to clip the original image into under and over segmented areas.
12. These areas of then resegmented using the same process in steps 1-3
13. New limits are utilized starting from the optimal scene wide segmentation scale
14. Data is read into R and merged back with the 80% unchanged segments were Moran I and normalized variance are again computed
15. The final segmentation scale including the three scales are used to segment the entire scene

### 3.1 Segmenting an Image: (Orfeo Toolbox)

**Step 1:** in OSgeo4W command line type otbGui_segmentation to open the segmentation window (Figure )**Error! Reference source not found.**.

**Step 2:** Click the 

button and navigate to the location of your data.

**Step 3:** Set your desired values for spatial radius, range radius, and minimum size (Figure 12). These can be determined by the process and scripts provided, but ultimately should be tuned by human evaluation of the final segmentation.

**Step 4:** Set the processing mode: This can be either a raster output as shown below or a vector output. The raster output is more memory intensive and was only applicable to Jamaica Bay for this study.

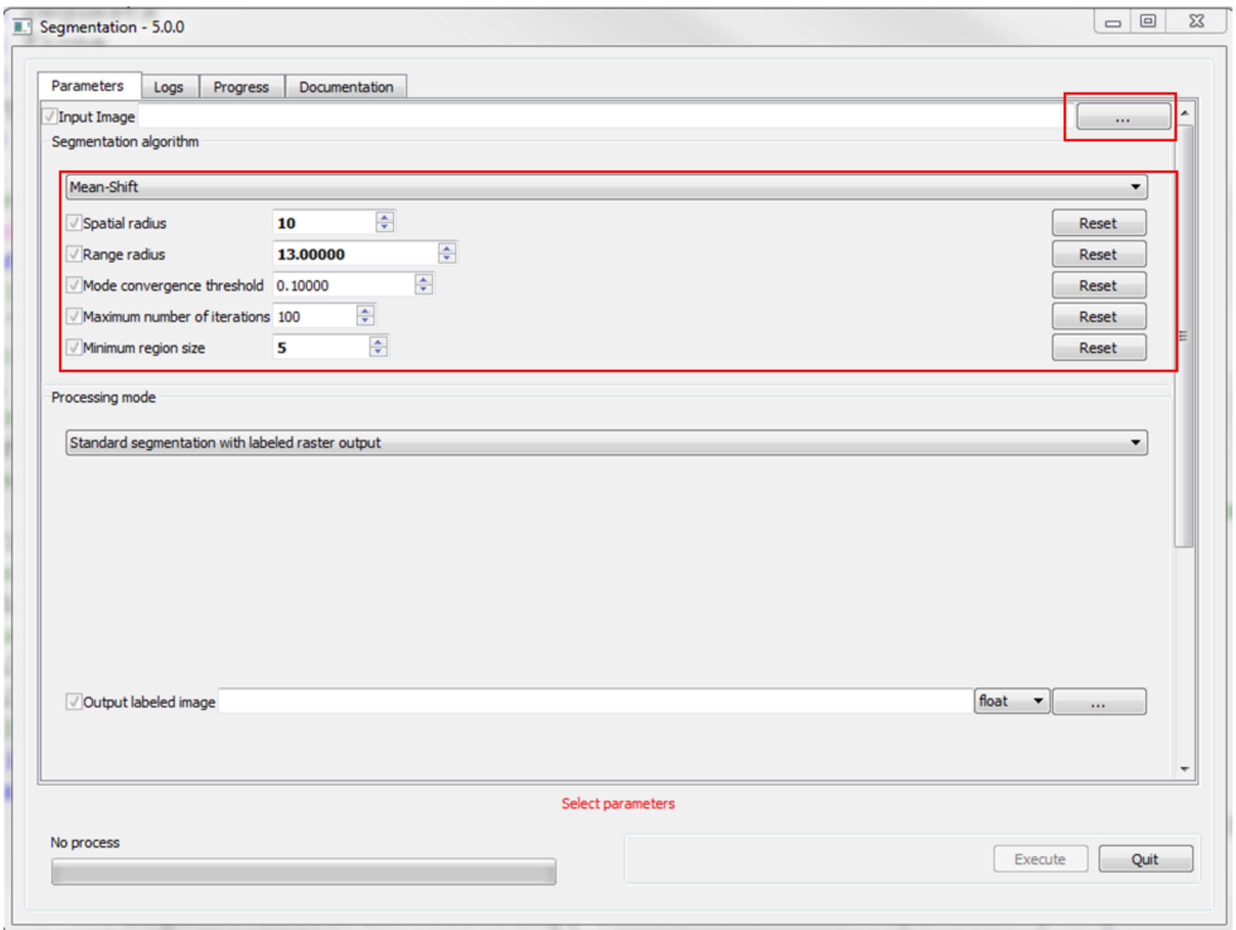



**Figure 12.**The segmentation window of Orfeo Toolbox. Change the input image, Range Radius, Minimum region size, and spatial radius. These variables determine the segmentation output which will be a raster in this example.

If using the raster output mode, the final step after the image has been segmented is to convert the raster to polygon data. This can be accomplished with the gdal_polygonize command within OSgeo4W.

**Step 1:** Change directory

Cd /d D:\Radiocorrect\Testing_segments\post_test

**Step 2:** Run command converting raster to polygons

gdal_polygonize –f “ESRI Shapefile” –b 1 pst_seg_5_15.img pst_seg_5_15.shp

This data processing step can be done with a variety of software, many other software packages take a sizeable time processing the conversion with a great number of unique objects and cannot handle the data type output by the segmentation.

# 4. Calculating Parameters

OBIA classifications use a variety of data parameters including those derived from base data such as the spectral bands, vegetation indices, and texture. LiDAR-derived elevation is often incorporated. Additional imagery dates can be incorporated similarly to the base data. Methods for creating DEMs can be found in many places or downloaded from the data provider. Vegetation indices are spectrally derived indices which can be created for a particular sensor (i.e. WVWI) bands or applied across most earth observation satellites (i.e. NDVI). Texture can be calculated in many ways and is another method for providing greater geospatial context to a classification. Texture measures take into consideration a pixels neighborhood, i.e. those pixels surrounding a central pixel, the neighborhood size must be defined to calculate texture.

### 4.1 Vegetation Indices (ERDAS Imagine)

Multiple vegetation indices should be calculated including vegetation and water focused. Recommended indices include Normalized Difference Vegetation Index (NDVI), WV-2 Vegetation Index (WVVI), WV-2 Water Index (WVWI) and Soil Adjusted Vegetation Index (SAVI) (Table 2). The pansharpened images were used, which in conjunction with vegetation indices have been shown improve the discernment of small vegetation patches (Johnson, 2014).

**Table 2.** Several common vegetation index equations with possible use in classifying salt marsh vegetation.

| WVVI | WVWI | Red Edge Vegetation Index | NDVI | SAVI |
|---|---|---|---|---|
| $\frac{(NIR2 - Red)}{(NIR2 + Red)}$ | $\frac{(CB - NIR2)}{(CB + NIR2)}$ | $\frac{(NIR1 - Red\ Edge)}{(NIR1 + Red\ Edge)}$ | $\frac{(NIR1 - Red)}{(NIR1 + Red)}$ | *(NIR – RED) *(1 + L) / (NIR + RED + L)* |

The equations are included within ERDAS Imagine except red edge vegetation index which can be calculated by selecting NDVI and changing the red component to red edge (band six instead of five) Figure 13). Calculate indices can be found in the Raster menu -> unsupervised -> Indices (Figure 14).

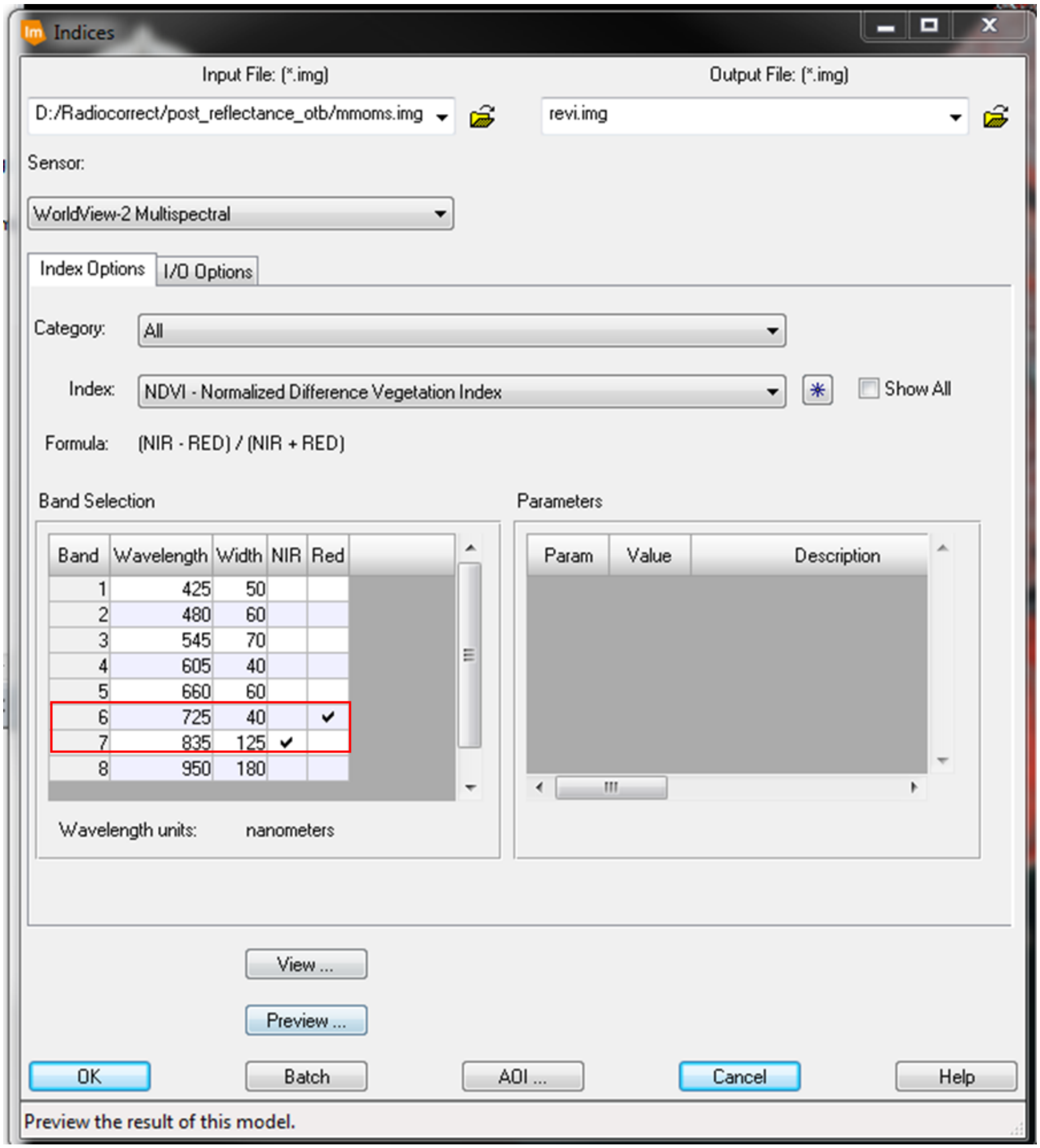



**Figure 13.** The NDVI index altered to calculate red edge vegetation index.

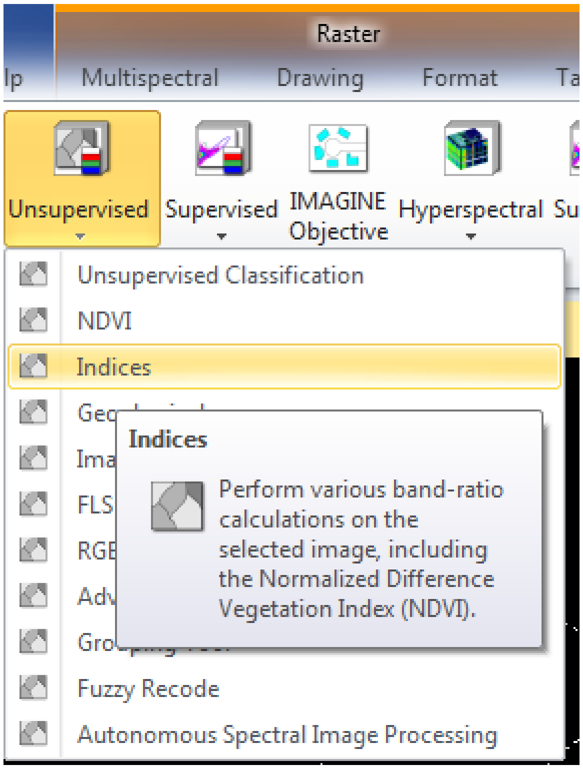



**Figure 14.** The location of the indices command in ERDAS Imagine, used to calculate spectral indices.

**Step 1:** Select the input .img file using the folder or drop down menu (Figure 15).


**Figure 15.** Input data for the indices tool.

**Step 2:** Provide a location and name for the output file (Figure 16).

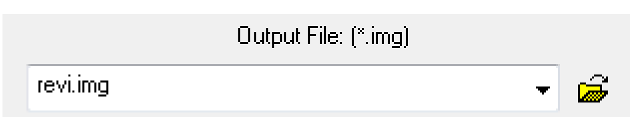



**Figure 16.** Output file for the vegetation index.

**Step 3:** Select the index of interest from the Index drop down window (Figure 17).

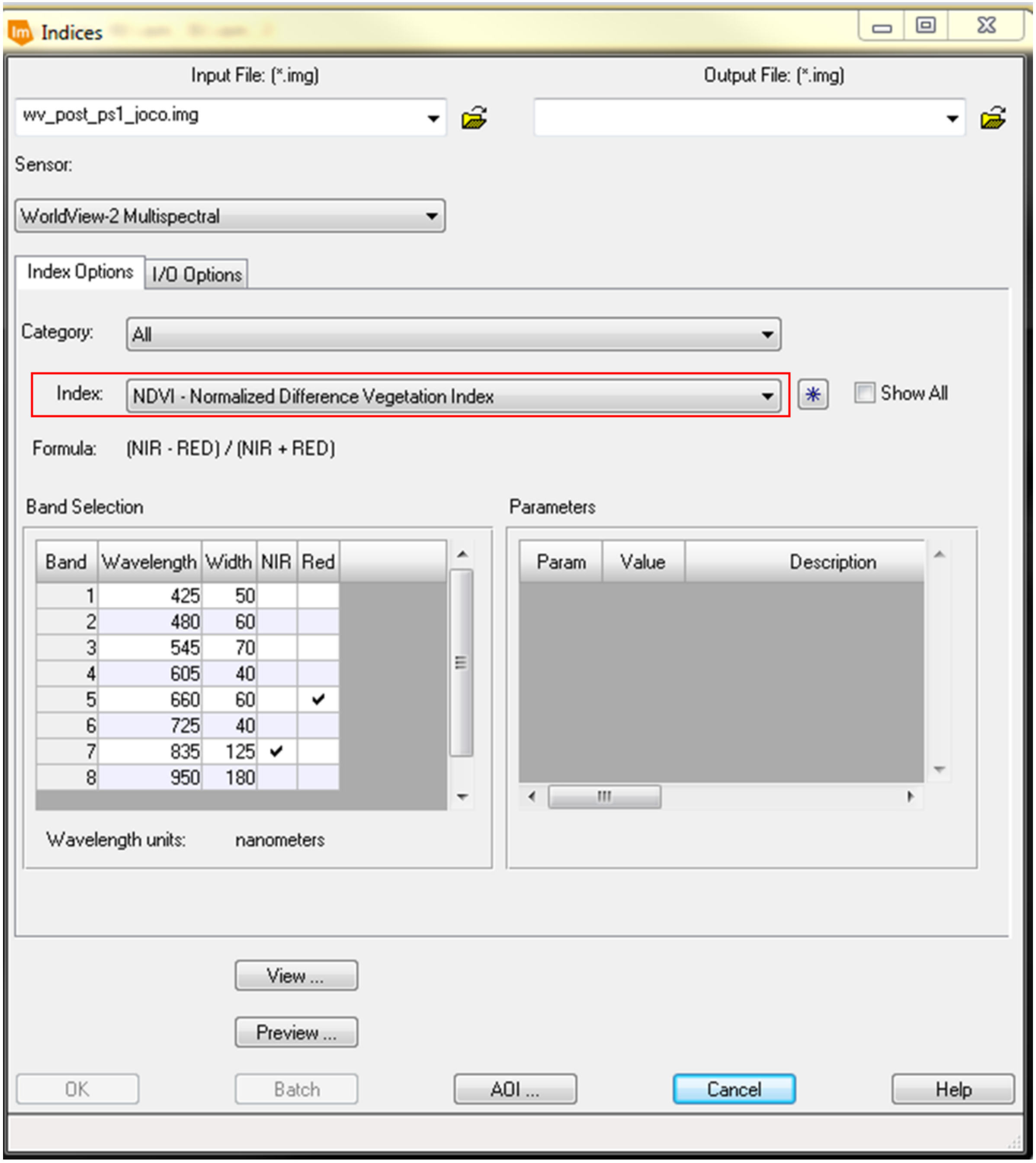



**Figure 17.** Calculating Vegetation Indices in ERDAS Imagine.

**Step 4:** Repeat this process for all indices SAVI, NDVI, WVWI, WVVI, and Red Edge Vegetation Index.

### 4.2 Texture: (R Statistical Software)

Grey level co-occurrence matrix is one method to calculate texture. GLCM has been shown to improve classification accuracies (Akar & Gungor, 2014). These measures are calculated with the panchromatic band as it represents our highest resolution spectral data. Texture will be calculated with the R package ‘glcm’ (Figure 18). The working directory and raster name should be changed when adapting this to a new set of data (lines 6-7 in the script).

```
1  install.packages('glcm')
2  install.packages('raster')
3  library(raster)
4  library(glcm)
5  #Change to working directory, location of the panchromatic dataset
6  setwd('D:\\Radiocorrect\\Pre_QB_refl_otb')
7  pan <- raster('preqbps1.img')
8  grey <- glcm(pan,n_grey=32,window=c(7,7), shift=c(1,1),statistics=c('homogeneity',
9                                   'contrast','dissimilarity','entropy','second_moment','correlation'))
10 writeRaster((grey,'texture.img',format='HFA''))
```

**Figure 18.** Grey level co-occurrence matrix code block. The raster written out from the script will be the six texture measurements.

# 5. Parameterization

SAGA GIS was found to be the quickest at performing statistics for polygons and as such was utilized to parametrize our image segments (Figure 19). Saga GIS is another component available with an installation of OSGeo4W. GIS data both raster and vector must be opened with the open command in Saga GIS. The generated texture, vegetation indices, spectral bands, and ancillary data will be used to calculate statistics for each of our image objects. The final parameters can be generated directly from the image objects which are shape indices, perimeter, and area. When additional data are available such as the QB-2 imagery from Jamaica Bay these can be added during the parameterization. Assateague Island included NAIP imagery and Sentinel-2 imagery and was done in QGIS. The classification of FIIS used Sentinel-2 and NAIP imagery too.

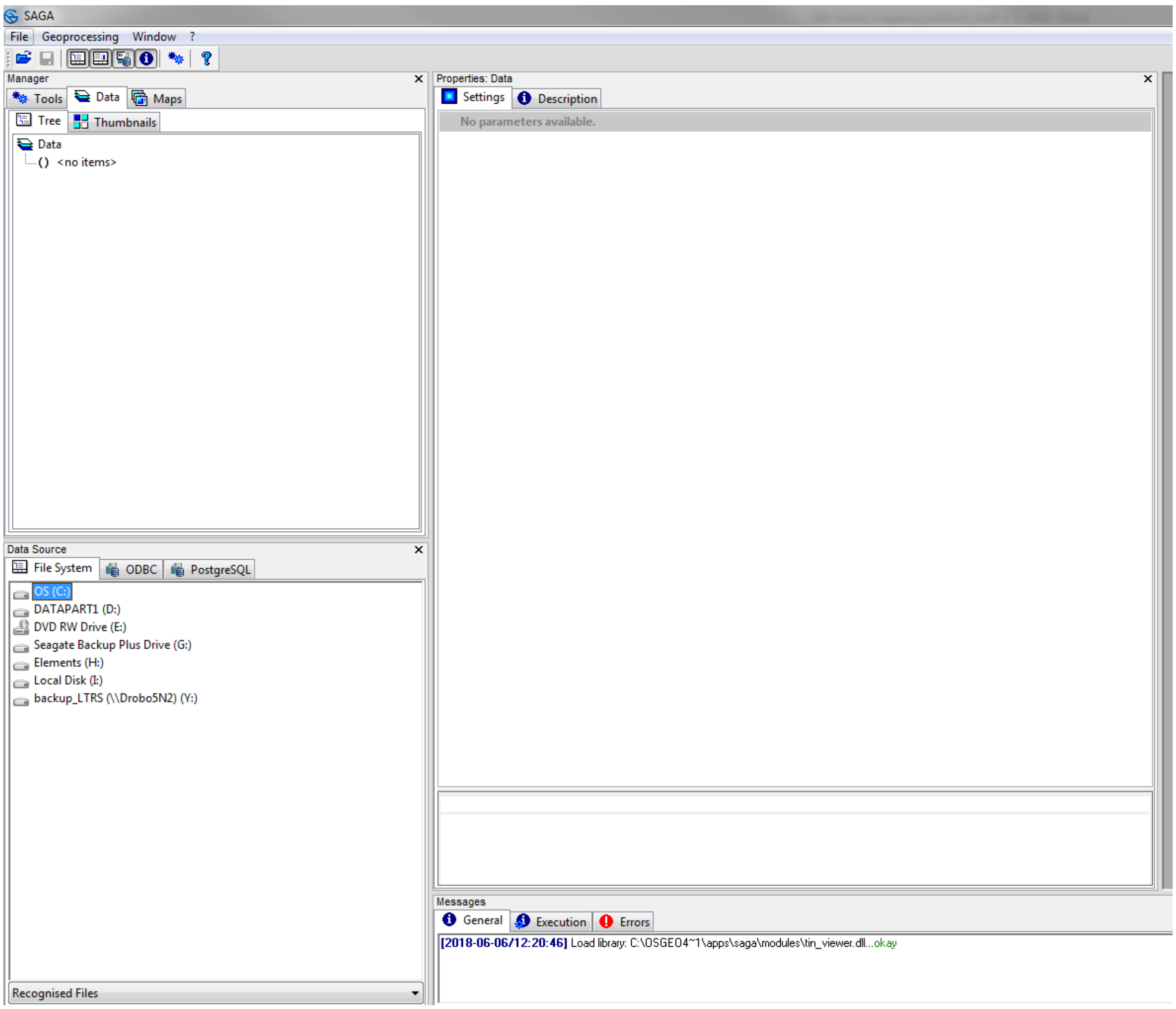



**Figure 19.** SAGA GIS - data can be added with the File menu. The tools used in the workflow can be accessed in the Tools tab.

**Step 1:** Open vector and raster data including imagery, texture and DEM. This can be done with the file open symbol (Figure 20 in the red box) or File -> Open.


**Figure 20.** Opening data in SAGA GIS.

Next an explorer window will open to select the files of interest. To reveal all file formats including (.img) select All Files from the right bottom corner (Figure 21).

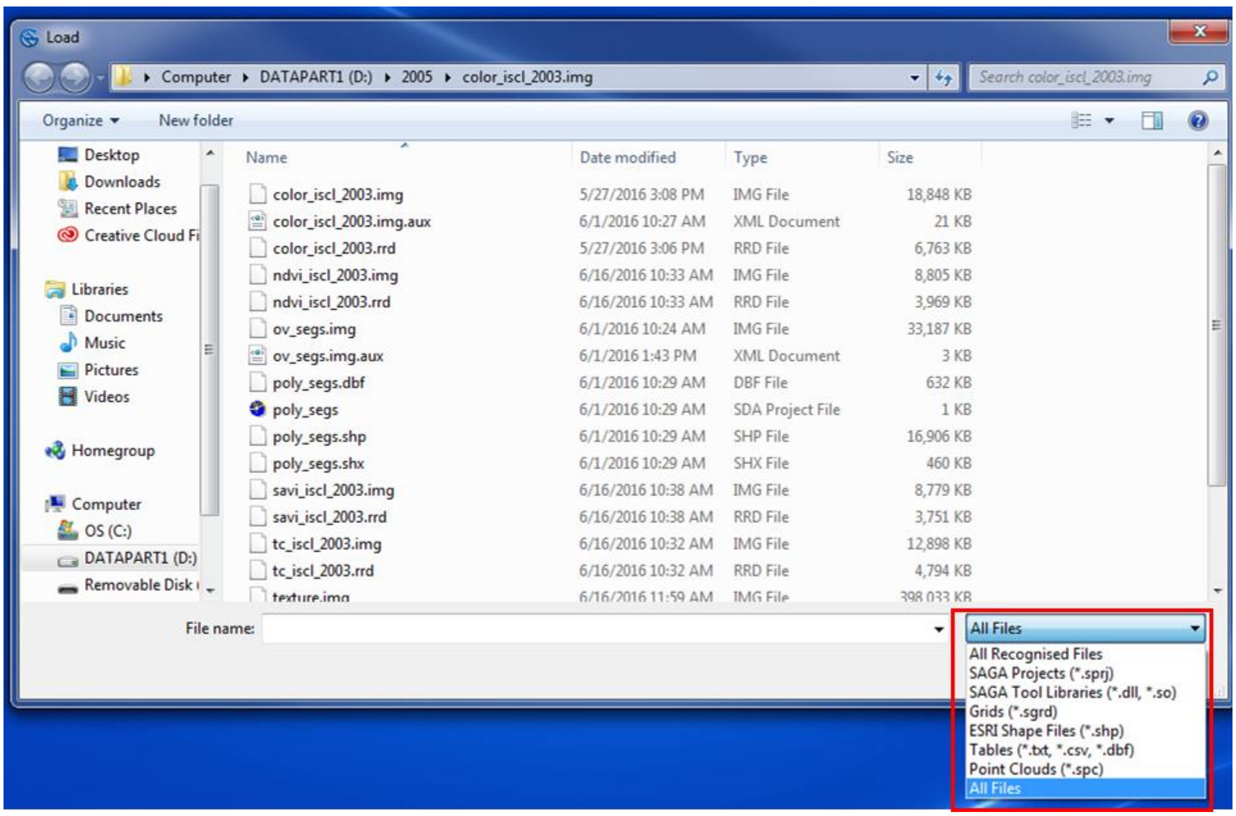



**Figure 21.** The SAGA GIS load file window.

Once a raster of interest is select the raster import dialog will open and the load all bands option should be check (Figure 22).

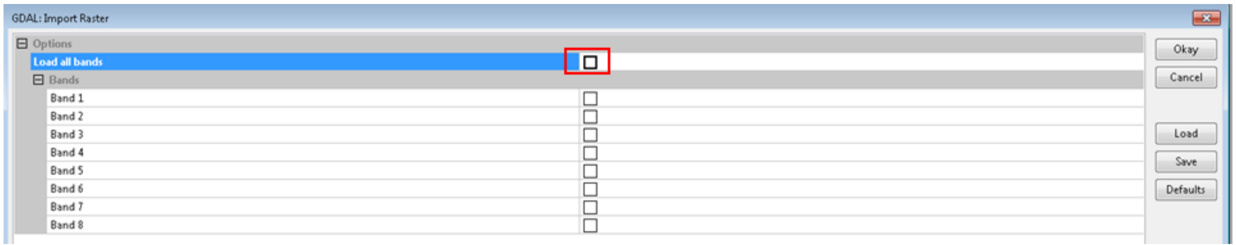



**Figure 22.** Loading raster data in SAGA GIS.

**Step 2:** The raster band names should be changed to more accurately represent the data. In this study we named each band after it's abbreviation for WV-2 data i.e. CB, B, G, Y, R, RE, NIR1, NIR2. The QB-2 data were named qB, qG, qR, qN. Texture was abbreviated as three letter abbreviations.

Double click a band demonstrated in the red box to open setting. The name of the band can then be changed. The coastal blue band has already been changed to CB (Figure 23).

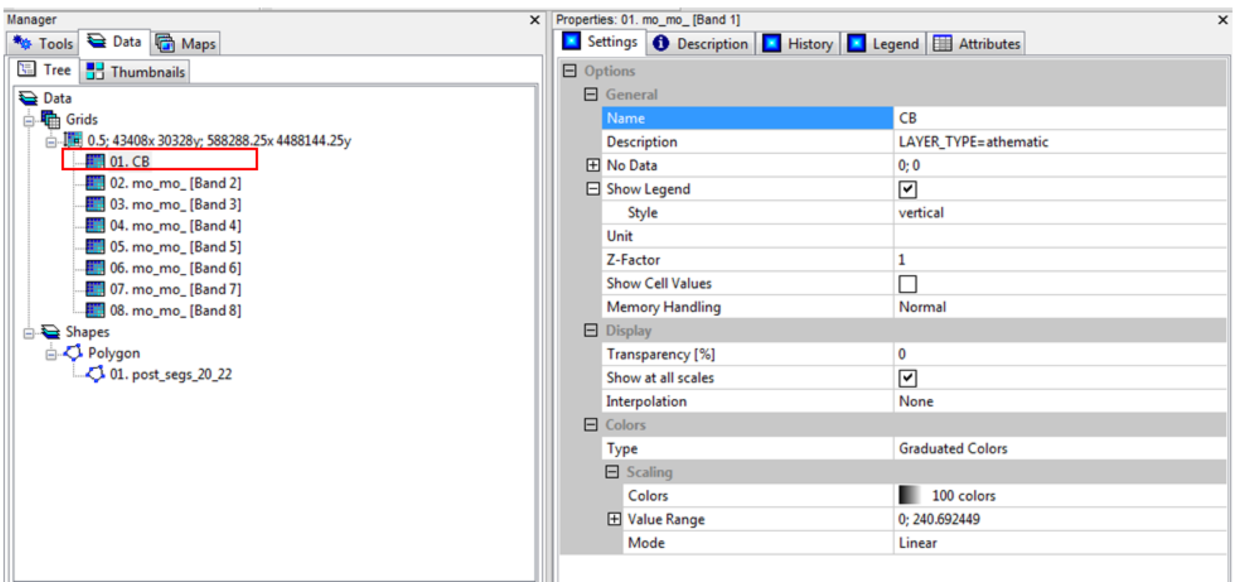



**Figure 23.** Renaming the raster bands in SAGA GIS.

## 5.1 Geospatial Attributes (SAGA GIS)

Can be calculated within SAGA GIS with the Polygon Properties tool found in Tools -> Shapes-> Polygons -> Polygon Properties. The perimeter, area and # of vertices were calculated with this tool.

## 5.2 Parametrizing Image Objects (SAGA GIS)

For this analysis the spectral parameters, textural, and indices attributes are calculated with SAGA GIS. The SAGA tool utilized is the within Tools -> Shapes ->Grid -> Grid Statistics for Polygons (Figure 24).

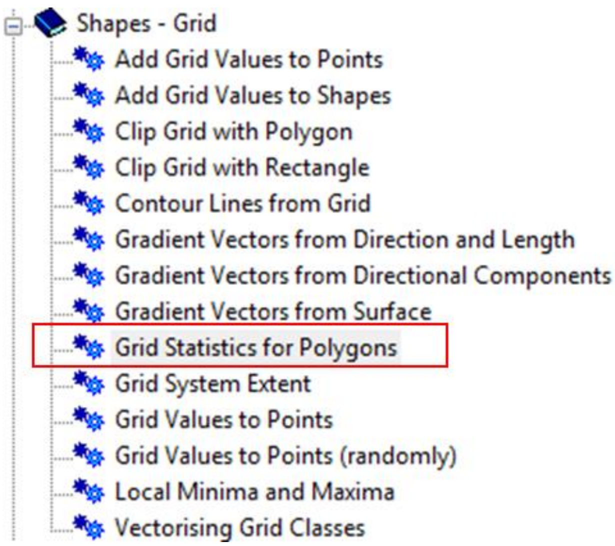



**Figure 24.** Grid Statistics for Polygons function.

Mean and standard deviation for all grids can be computed with the Grid statistics for Polygons tool. This includes the GLCM texture maps created, spectral data and ancillary data such as Digital Elevation Models.

**Step 1:** Click the down arrow next to the grid system and select an available grid (Figure 25).

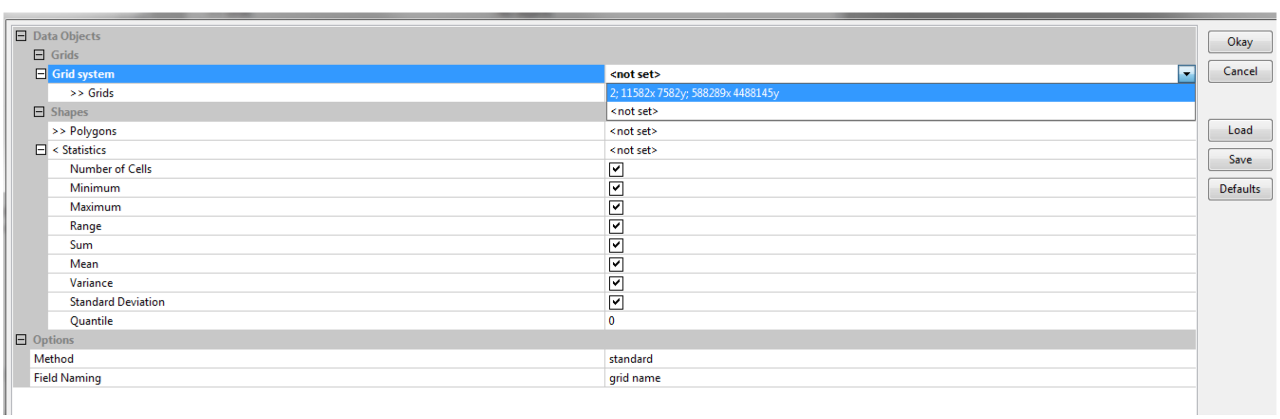



**Figure 25.** Grid system selection in Grid statistics for polygons.

**Step 2:** Next select click the  and add all grids that correspond with the system (Figure 26).

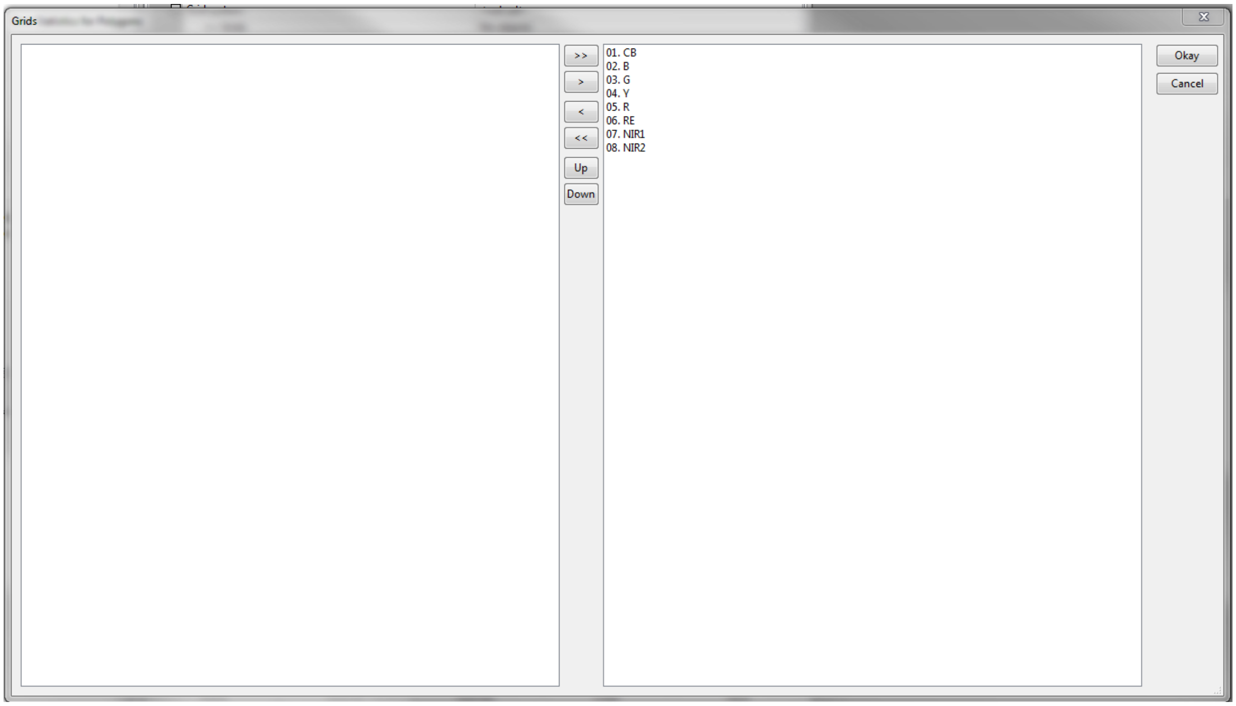



**Figure 26.** The grid selection window of the Grid statistics for polygons tool.

**Step 3:** Identify the polygon data to use in the tool. These must be opened using the file menu. Under the shapes -> Polygons heading select your segmentation. Select the same segmentation in the statistics heading (Figure 27).

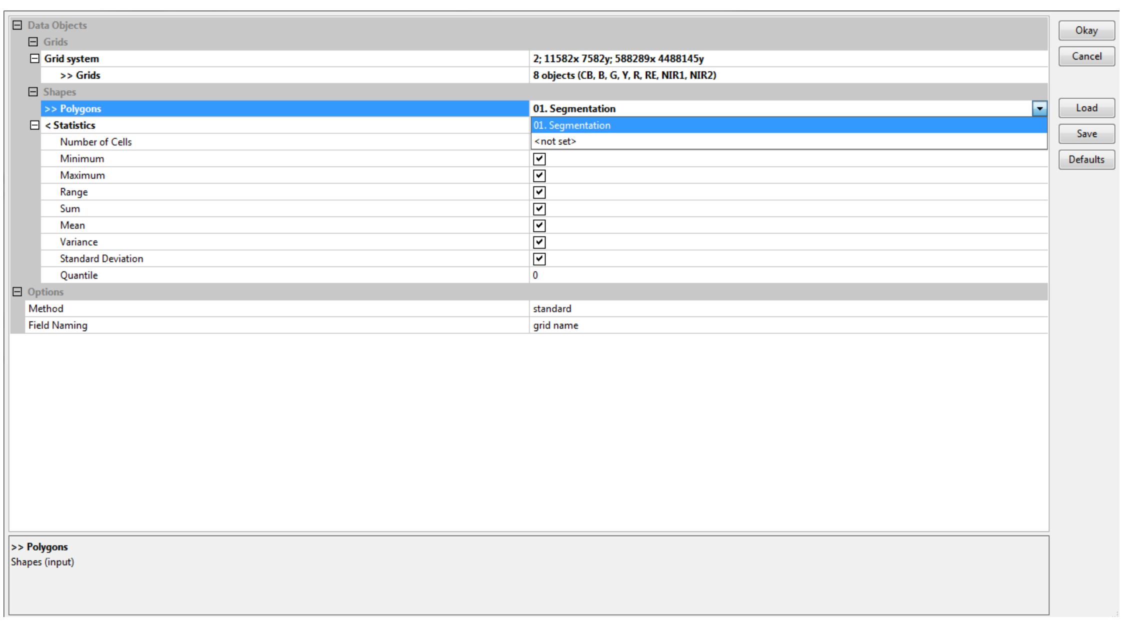



**Figure 27.** The grid statistics for polygons tool, showing the process of selecting your polygons.

**Step 4.** Select the statistics to run by checking or unchecking the corresponding box (Figure 28).

The example below shows the dialog and parameters to calculate mean and standard deviation for each of the spectral bands (Figure 28).

If a raster has the same grid system they can be calculated together, this should include the texture and spectral attributes. Read this process for each set of grids.

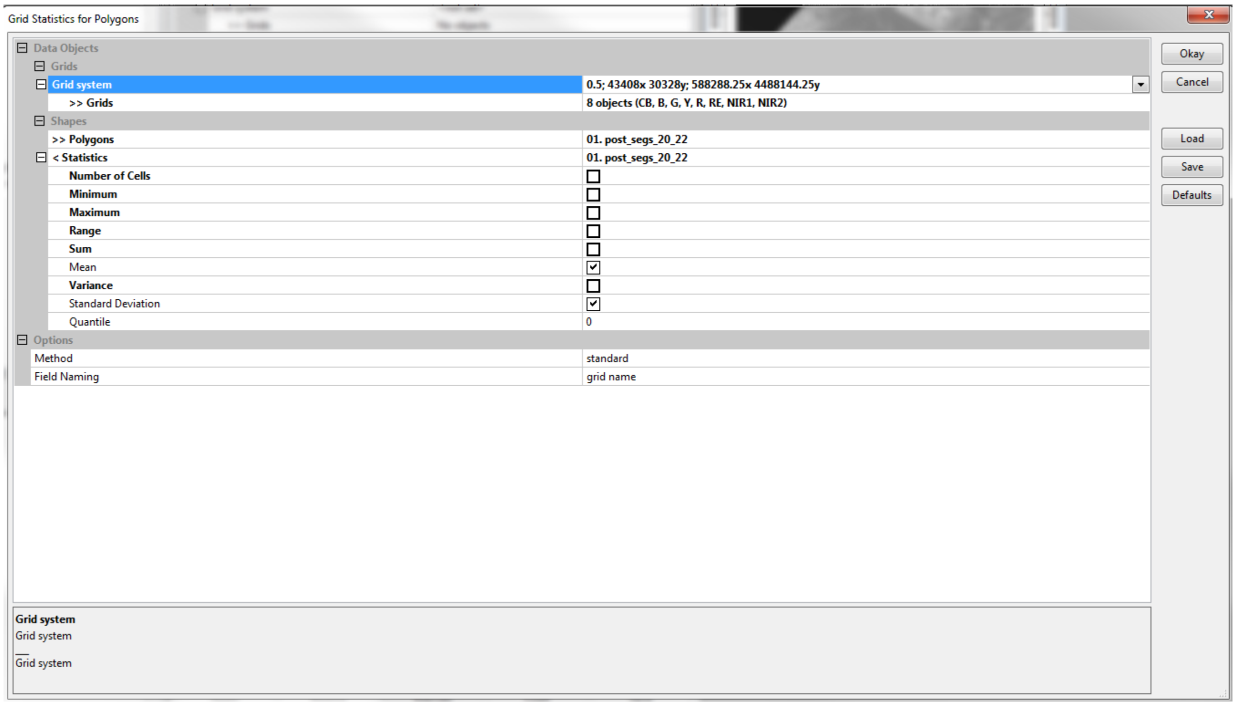



**Figure 28.** Calculating grid statistics for the image objects.

**Step 2:** Repeat the process for any other data sets being used such as a second date of imagery for this study the QB-2 data for the same year and season was also calculated for the WV-2 image objects.

## 5.3 Parametrizing Image Objects (QGIS)

The parametrization of ASIS image objects was conducted with QGIS due to the size of the rasters being used for the site. An example of the process for ASIS with sentinel-2 data is detailed below. First the band names are defined for the output columns then the necessary functions are imported (Figure 29, lines 3-5). The image objects are defined (Figure 29, line 6) and rasters these lines should be updated with new file locations (Figure 29, 7-8). Next the layers are checked for validity (Figure 29, lines 9-11). Mean and standard deviations are calculated for each object and raster band (Figure 29, Line 14-18). Next mean and standard deviation are calculated for the NDVI layer. The vector layer will have new fields named after the band names and statistic (Figure 30).

```
1  bands= ["NA","S1","S2","S3","S4"]
2  name1="Sndvi"
3  from qgis.core import QgsVectorLayer
4  from qgis.core import QgsRasterLayer
5  from qgis.core import *
6  vLayer = QgsVectorLayer("D:\\ASIS\\NAIP_2015\\mid_naip\\NP_MID.shp","MD_12s","ogr")
7  rasterpath="D:/Sentinel/s2a_mo.img"
8  rasterpath1="D:/Sentinel/s2a_mondvi.img"
9  rlayer=QgsRasterLayer(rasterpath,"zone_1")
10 if not (vLayer.isValid() and rlayer.isValid()):
11     print "Error loading layers..."
12 from qgis.analysis import QgsZonalStatistics
13 for i in range(1,5):
14     zonalstats = QgsZonalStatistics(vLayer,rasterpath, bands[i],i,QgsZonalStatistics.Mean)
15     zonalstats.calculateStatistics(None)
16 for i in range(1,5):
17     zonalstats = QgsZonalStatistics(vLayer,rasterpath, bands[i],i,QgsZonalStatistics.StDev)
18     zonalstats.calculateStatistics(None)
19 zonalstats = QgsZonalStatistics(vLayer,rasterpath1, name1,1,QgsZonalStatistics.Mean)
20 zonalstats.calculateStatistics(None)
21 zonalstats = QgsZonalStatistics(vLayer,rasterpath1, name1,1,QgsZonalStatistics.StDev)
22 zonalstats.calculateStatistics(None)
```

**Figure 29.** Python script calculating image object parameters for ASIS from Sentinel-2 data.

| | S1mean | S1stdev |
|---|---|---|
| 2 | 1064.000000000... | 54.62600113499... |
| 3 | 1092.000000000... | 0.000000000000... |
| 4 | 975.8181818181... | 77.22097197262... |
| 5 | 1104.000000000... | 0.000000000000... |
| 6 | 1104.000000000... | 0.000000000000... |
| 7 | 880.0625000000... | 2.948119103767... |
| 8 | 874.0000000000... | 0.000000000000... |
| 9 | 873.3043478260... | 5.188324831365... |

**Figure 30.** The created columns within the image object shapefile for band 1 of the sentinel-2 data.

## 6. Neighborhoods (R Statistical Software)

Next the data should be imported into R using the provided script (Figure 31). The script calculates neighborhoods for each object and the difference between an image object and the average of its neighborhood and scene. The neighborhood is defined as all the objects which a single object directly touches. These variables are calculated for the means of all spectral and textural data (Figure 31).

```r
#Install Required packages
install.packages('rgdal')
install.packages('spdep')
#Load required packages
library(spdep)
library(rgdal)

##Set the work directory where the data resides
setwd("D:/FIS/all_5_13")
#Read in the shapefile
test1 <- readOGR(getwd(),"FIIS_2scale2")
#Remove unneccessary characters from column names
names(test1) <- gsub("[.]", "",names(test1))
#define a polygon around each object to use in the neighborhood definitions
polbox <- poly_findInBoxGEOS((test1))
testnbs <- poly2nb(as(test1,"SpatialPolygons"),foundInBox = polbox)
#Create neighborhood object from the shapefile
#If any characters read in as . remove those
names(test1) <- gsub("[_]", "",names(test1))
#create neighborhood weights from the neighborhoods, all neighbors are given equal weight
testwts <- nb2listw(testnbs, style="W", zero.policy = TRUE)
#Calculate the mean for the neighborhood
NSUM1<-apply(test1@data[1:length(test1@data)],2,function(x) lag(testwts,x, zero.policy=TRUE))
#Convert to data frame
NSUM1=as.data.frame(NSUM1)
#Calculate scene wide mean
Means1 <- apply(test1@data[1:length(test1@data)],2,function(x) mean(x))
#Determine neighborhood difference
Ndiff1 <- mapply(function(x,y) {x-y},test1@data[1:length(test1@data)],NSUM1)
#Calculate neighborhood and scene difference attributes
for (i in 1:length(Means1)) {test1@data[paste(substr(colnames(test1@data[i]),1,6),"Ndi",sep="")]=test1@data[i]-NSUM1[i]}
for (i in 1:length(Means1)) {test1@data[paste(substr(colnames(test1@data[i]),1,6),"Sdi", sep="")]=test1@data[i]-Means1[i]}
writeOGR(test1,getwd(),'fiis_scene','ESRI Shapefile')
```

**Figure 31.** Code chunk also available with this workflow as a standalone script.

# 7. Training Object Selection (QGIS)

Areas of confirmed land cover should be used to create training objects. Since *in situ* training objects were not available for Jamaica Bay, training objects were collected from source imagery and other images of the study area. Our field visit locations were used as a guide, but were not enough data on their own to create the entire training set. The other images included NAIP imagery from a similar time period, Google Earth historical aerial imagery to create appropriate training dataset imagery should be used from a similar time period as the satellite imagery. Each class should have 50-100 training objects selected. Training objects should be taken from across the scene not a single marsh island. For this study training objects were selected using ArcGIS. The segmentation objects were first imported into a geodatabase which allowed for quicker manipulation of the image objects.

## 7.1 Training Objects

**Step 1:** Use the browser panel to navigate to the imagery and double click to add to the QGIS mapping window. Repeat the process for your image objects (Figure 32).

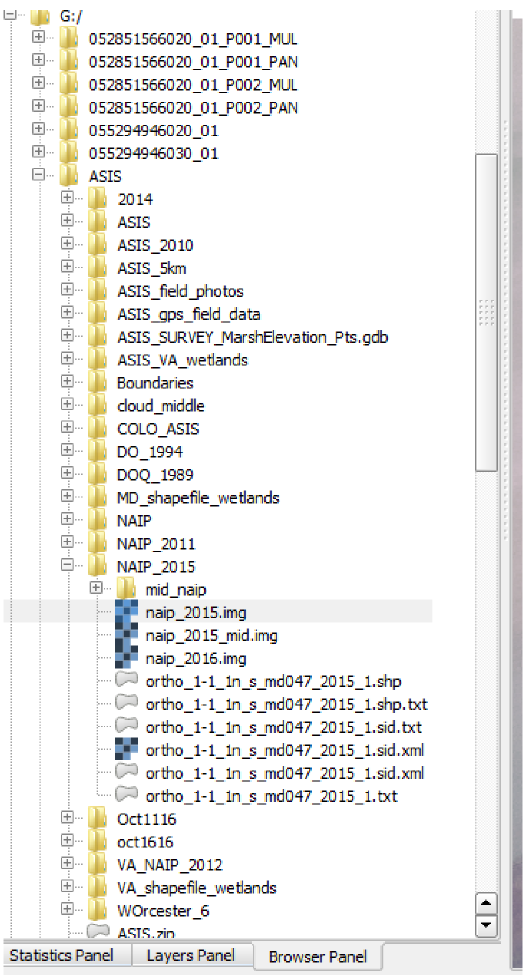



**Figure 32.** QGIS browser panel add data to QGIS using this file browser.

**Step 2:** Set the NAIP imagery to pseudo color. Open the layer properties by double left clicking the dataset in the layers panel. Click the drop down menu under red band and select band 4 (Figure 33).

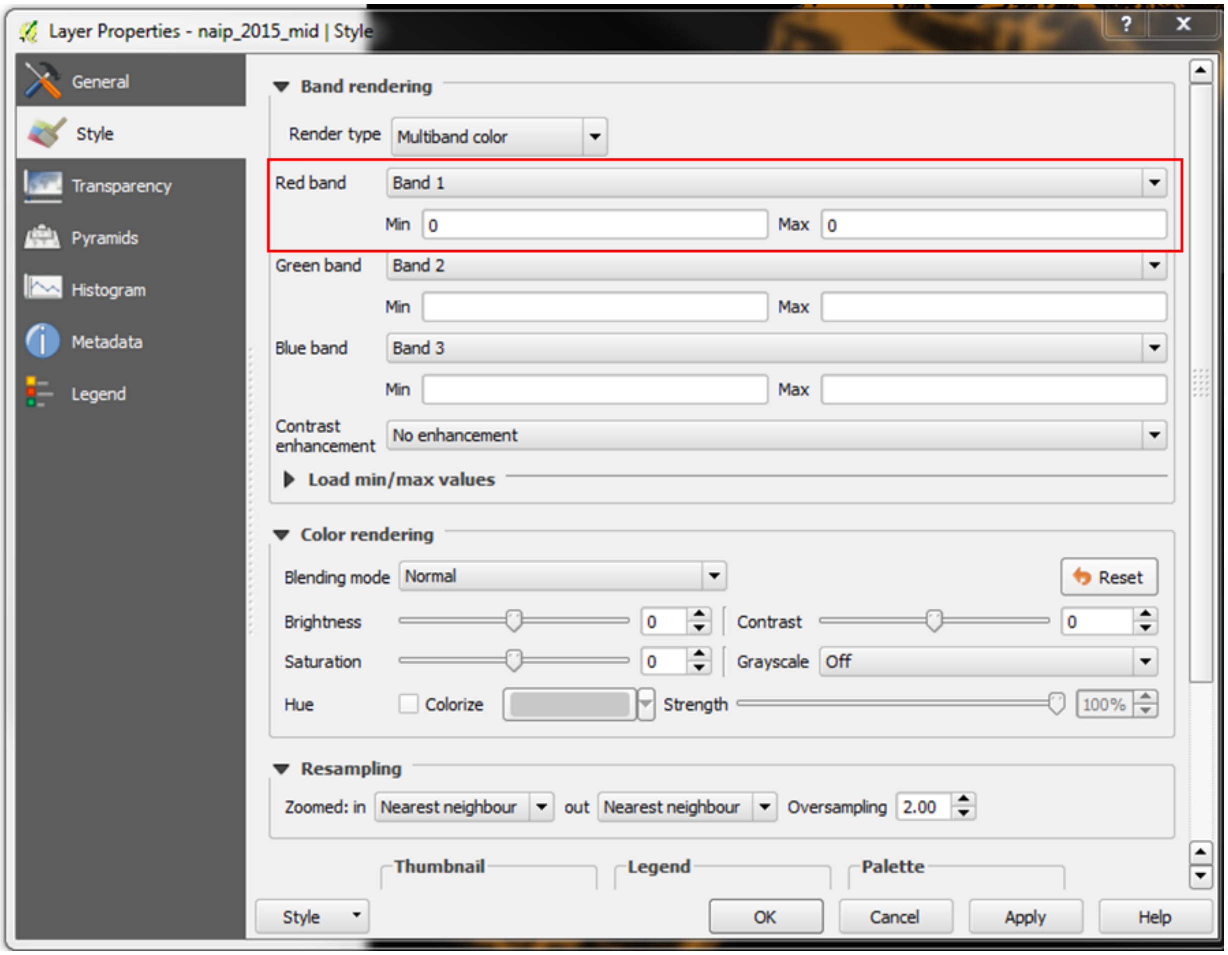



**Figure 33.** The QGIS Layer properties window for raster data.

**Step 3:** Calculate histogram and use min/max visualization. Select the histogram tab of the raster layer properties window. Click Compute Histogram (Figure 34).

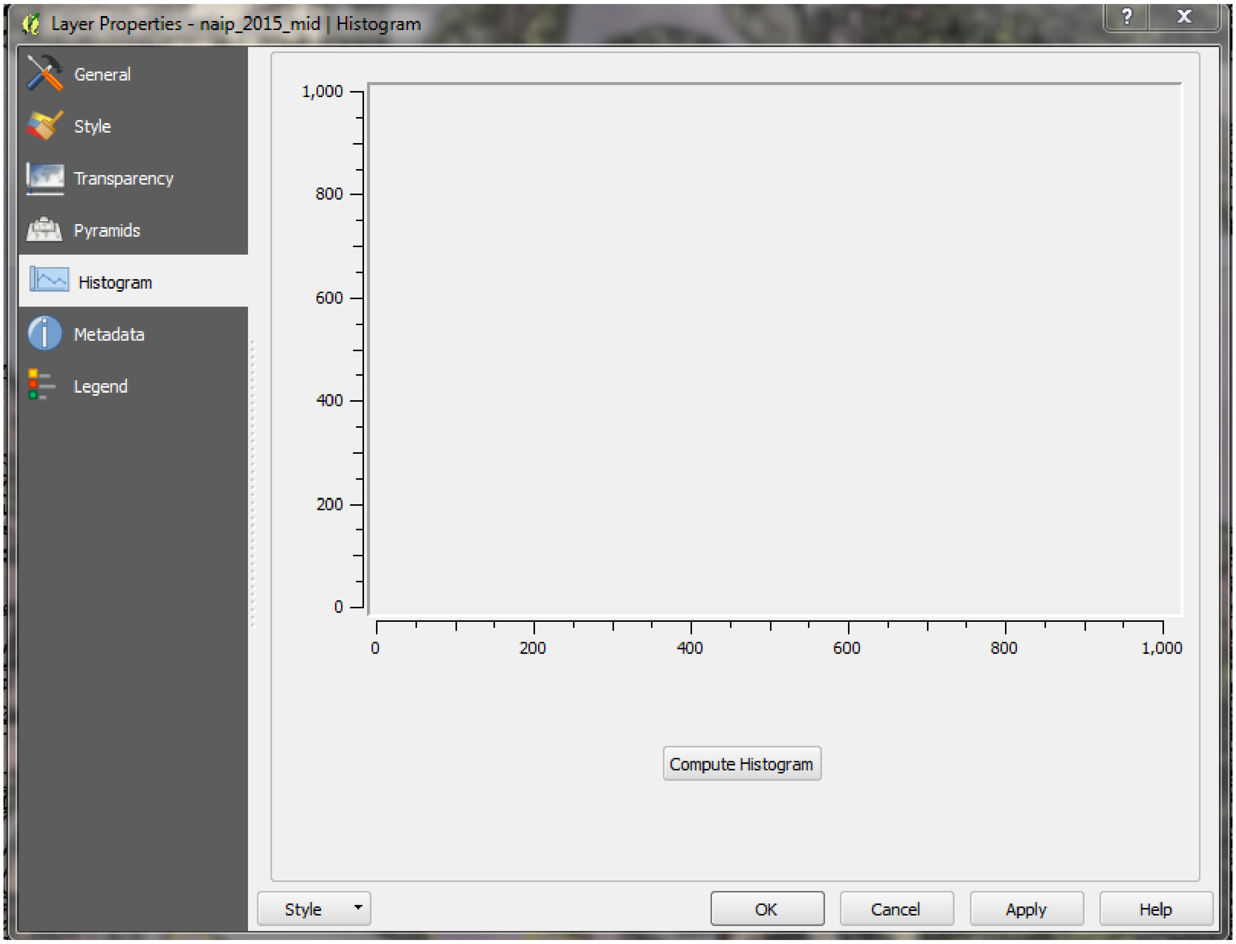



**Figure 34.**The histogram tab of the Layer properties window. Click compute histogram to calculate the image data histogram. This is important for image visualization.

**Step 4:** In the style tab of the Layer properties set contrast enhancement to Stretch to MinMax and open the Load min/max values tab and click load (Figure 35).

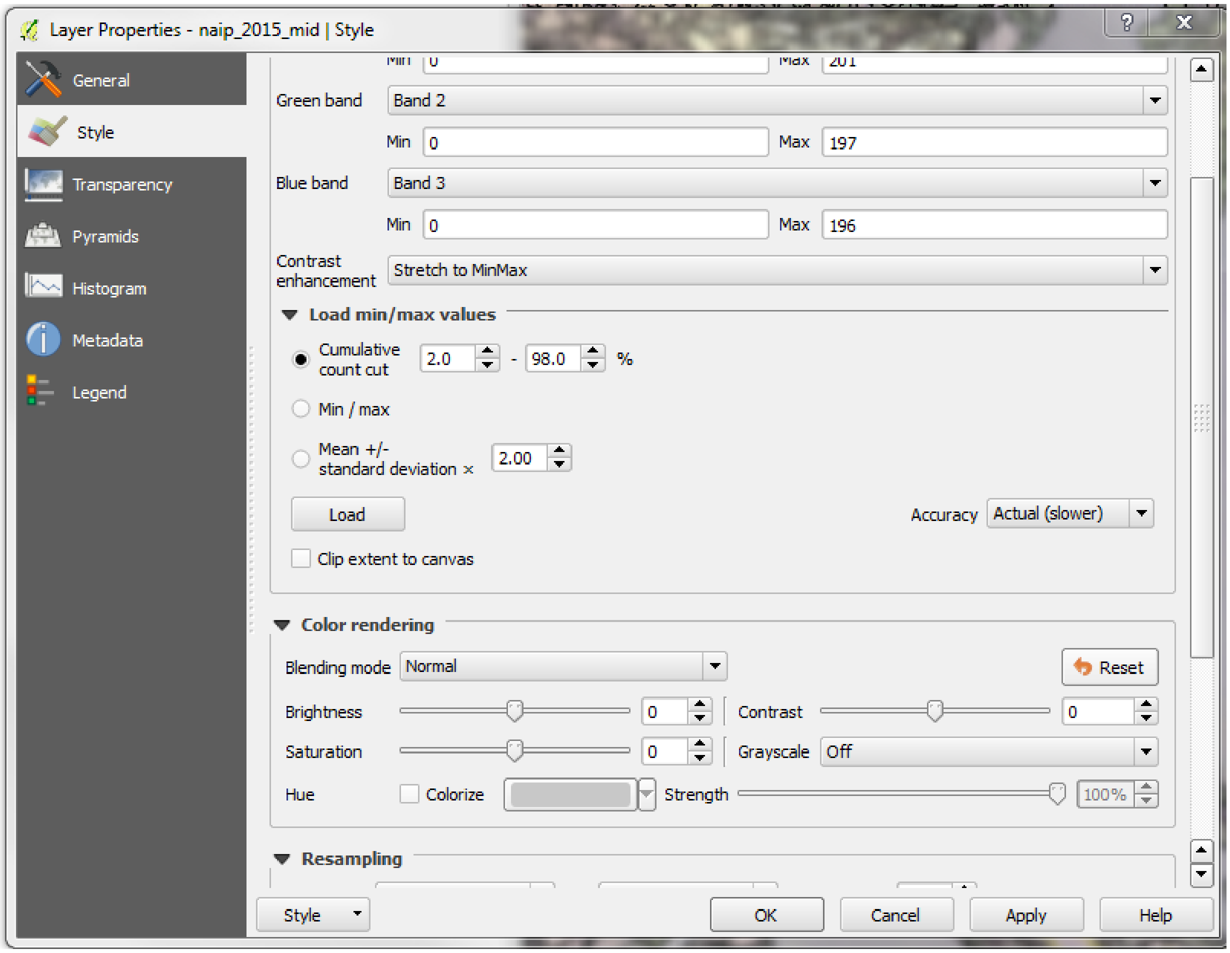



**Figure 35.** The QGIS Layer properties window. The contrast enhancement selection should be set to Stretch to MinMax.

**Step 5:** Set the image objects to display as hollow. Again double click the dataset and open the Layer Properties window (Figure 36).

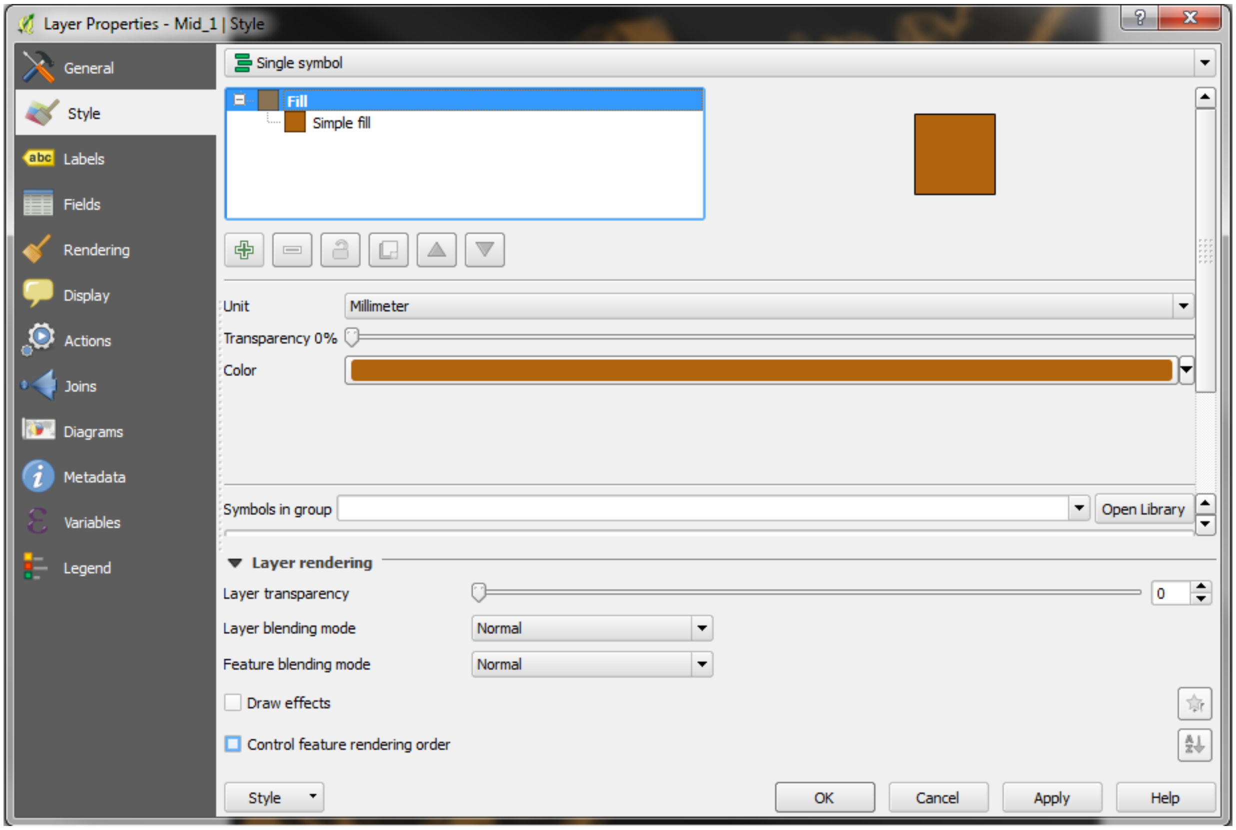



**Figure 36.** The QGIS Layer properties window for vector data.

**Step 6:** Select simple fill -> Symbol layer type dropdown menu and select outline simple line and click ok (Figure 37).

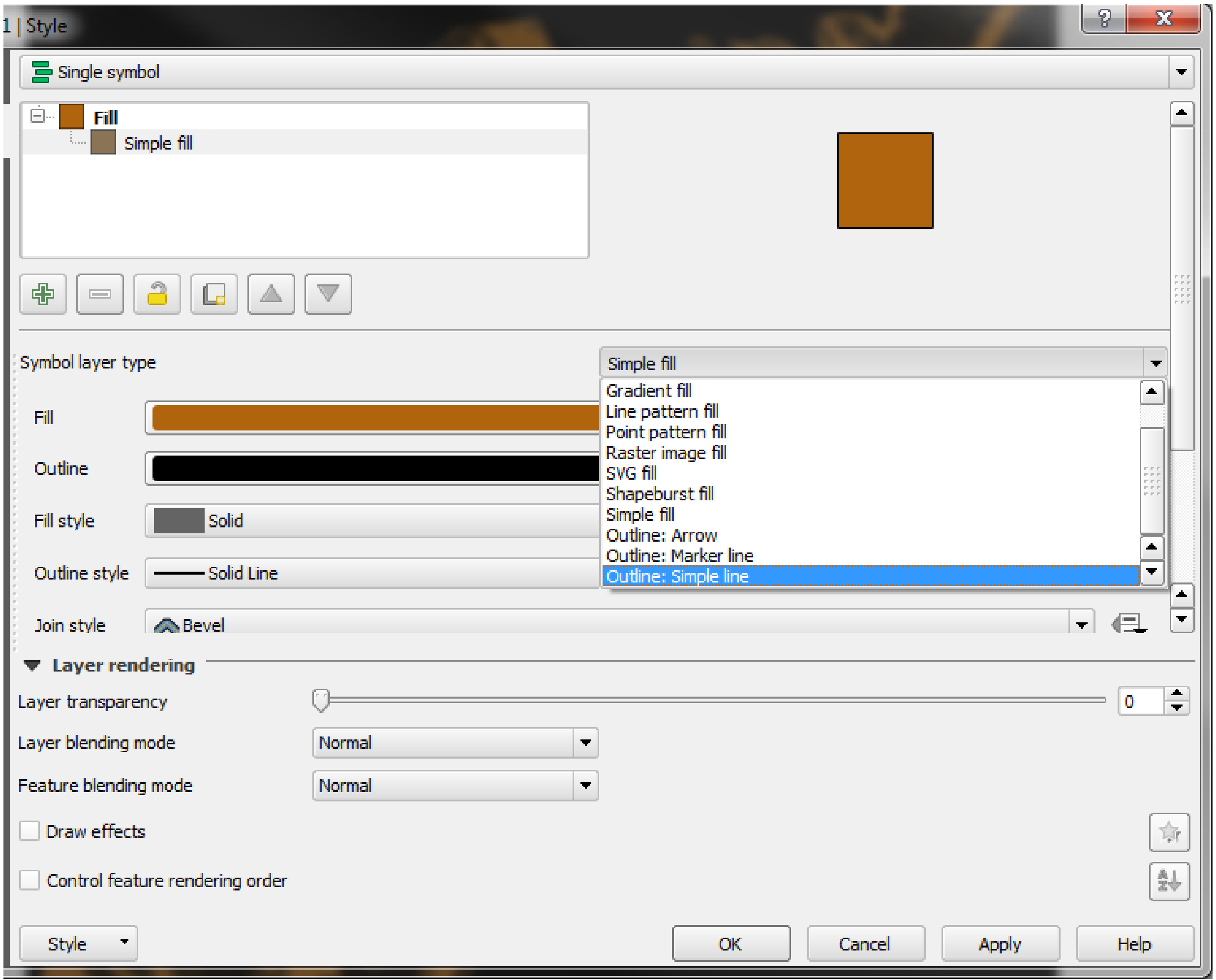



**Figure 37.** The process for creating hollow symbolization to facilitate the selection of training objects.

**Step 5:** Click the image objects in your layer panel. Use the selection  button to select objects.

**Step 6:** In the edit menu select Copy Features and then Paste Features as -> New Vector Layer (Figure 38).

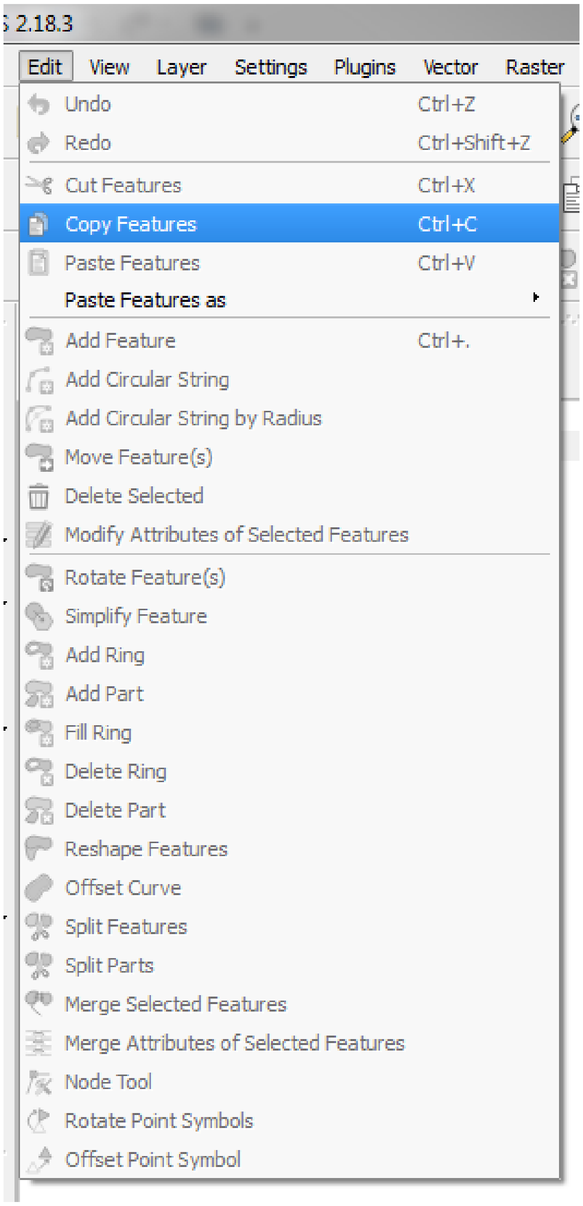



**Figure 38.** QGIS Edit menu options.

**Step 7:** Fill out the file name in the save vector layer as… Click browse and select the location and name of your training dataset (Figure 39).

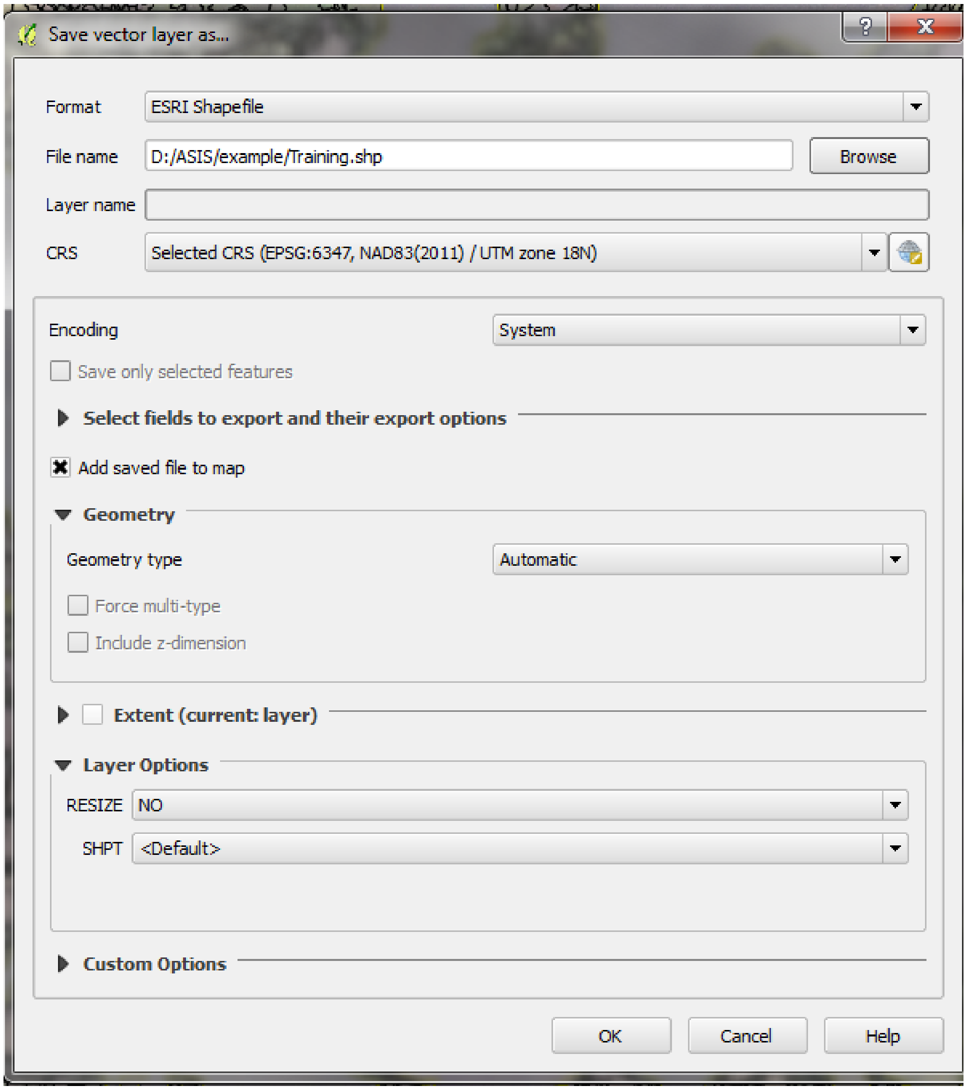



**Figure 39.** QGIS Save vector layer as... window.

**Step 8:** To add additional objects to the training data start editing the data by selecting it in the layers panel and clicking  button. Next select additional image objects and select copy features then paste features.

**Step 9:** Create and calculate class field. Click the  to open the Field calculator. Select create a new field, field name = Class, field type = Text(string). In the expression field add the class of the selected objects in single quotes (Figure 40). Add new image objects of a single class to your training set and calculate the Class field immediately following their addition.

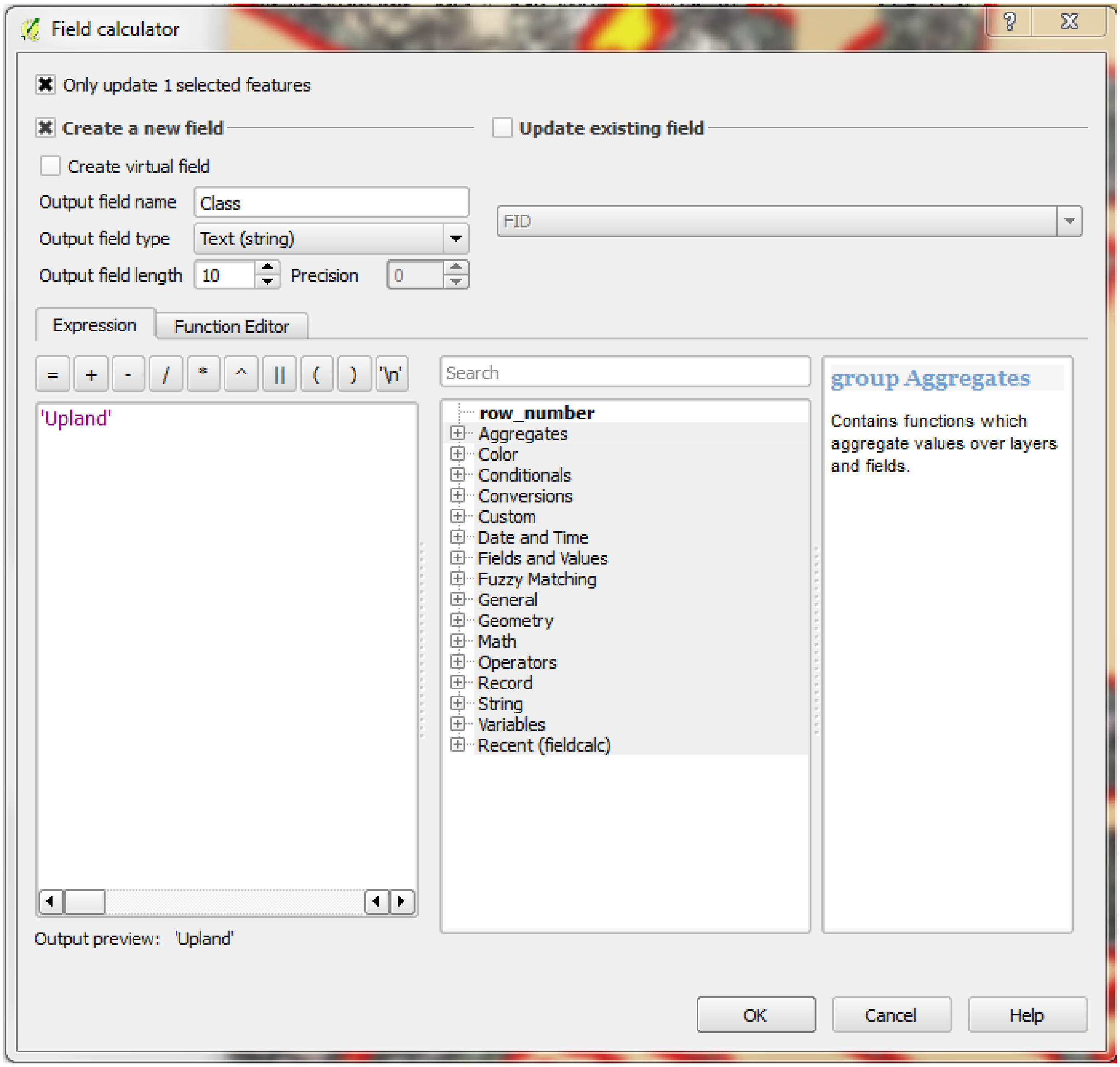



**Figure 40.** QGIS Field calculator used to calculate the Class field and give all selected objects the Upland class.

# 8. Machine Learning Classification

## 8.1 Overview

The classification process is rather simple within R. Once image objects have been parametrized and training objects selected either with training plot data or by the image analyst the data are read into R. Next unused training data fields are removed from the training data. Including fields that are unique to the training data such as ObjectID. Then the model is trained. The trained model is then used to classifying the data which at this point can be written out of R and analyzed or visualized with ArcGIS, QGIS or ERDAS Imagine.

## 8.2 Classification (R Statistical Software)

Classification is conducted with the Caret Package in R. The example script uses the Random Forest algorithm but a large variety of classification algorithms are available. The basic flow of the script is user sets working directory (Figure 41, line 8) where both training and image objects are located. Remove variables that should not be used from the training data for example OBJECTID (Figure 41, line 18). This step requires user input, but is only required for the training data. The classification is trained with the training data samples and then the data is classified (Figure 41, Line 23-24). The script example is for RF due to RF having the best performance in this study. The final output will be your polygon data with a new column called pred that has the predicted classes for each polygon.

The code for this step and the next two are included as a single file along with this document.

```r
install.packages('caret')
install.packages('rgdal')
install.packages('randomForest')
#Load packages
library(randomForest)
library(caret)
library(rgdal)
setwd("D:\\pre_5_15\\training") #set working directory
#Read in base data, data directory and then shapefile name after the comma without the extension
xvars <- readOGR(getwd(), "pre_refiner_5_20")
#Read in training data, data directory and then shapefile name after the comma without the extension
sdata <- readOGR(getwd(),"fresh_train_9")
#Convert to data frame remove spatial data for the classification
sdata.df <- as(sdata, "data.frame")
xvars.df <- as(xvars, "data.frame")
#Remove unnecessary attributes, this is just a few examples all other data that is no appliciable to the classification
#can be removed.
sdata.df$OBJECTID <- NULL
#Resampling objects created.
fitcontrol <- trainControl(method="repeatedcv", number=10, repeats = 10, savePredictions=TRUE)
#Random Forest example
rfGrid <- expand.grid(mtry=c(4,6,8,10,12,14))
rfFit <- train(Class ~., data=sdata.df, method="rf",trControl = fitcontrol, verbose=TRUE, tuneGrid=rfGrid)
rfPred <- predict(rfFit, newdata=xvars.df)
#Decision tree to remove shadow
xvars@data$RF <- rfPred
xvars@data$RF[which(xvars@data$RF == 'Shadow' & xvars@data$Upland ==1,)] <- 'Forest'
xvars@data$RF[which(xvars@data$RF == 'Shadow' & xvars@data$WVWIMEAN >.36)] <- 'Water'
xvars@data$RF[which(xvars@data$RF == 'Shadow' & xvars@data$WVVIMEAN >.38)] <- 'Spar50'
xvars@data$RF[which(xvars@data$RF == 'Shadow' & xvars@data$WVVIMEAN >.2)] <- 'Spar10'
xvars@data$RF[which(xvars@data$RF == 'Shadow')] <- 'Mudflat'
xvars@data$coords <- NULL
#Write a shapefile that includes the classified object categories
writeOGR(xvars,getwd(),'rf_object_classification','ESRI Shapefile')
#Create an index of training data based on the Class variable
trainIndex <- createDataPartition(sdata$Class, p=.60, list=FALSE,times=1)
#Divide the training data into training and testing data 60% as Training and 40% as Testing data
Train <- sdata@data[trainIndex,]
Test<- sdata@data[-trainIndex,]
#Next train the algorithms with the Training data
rfFit2 <- train(Class ~., data=Train, method="rf",trControl = fitcontrol, verbose=TRUE, tuneGrid=rfGrid)
#After training is conducted classify the testing data
Test$pred <- predict(rfFit2,newdata=Test)
#The shadow category should be removed from the testing data
test2nosh <- Test[-which(Test$Class == 'Shadow'),]
#Remove shadow category from both prediction and class categories
test2nosh$Class <- factor(test2nosh$Class)
test2nosh$pred <- factor(test2nosh$pred)
#Calculate confusion matrix for the testing data
confusionMatrix(test2nosh$pred,test2nosh$Class)
```

**Figure 41.** Code block demonstrating a classification using the Caret package in R statistical package.

### 8.3 Post-Classification (R Statistical Software)

A simple post classification is conducted in R to categorize the shadow category.

1. The tree first categorizes all shadow within the upland layer as upland vegetation
2. Next any shadow above a WVWI of .36 is defined as water
3. Any vegetation above .38 WVVI is defined as Spartina 50-100% density
4. Any vegetation above .18 WVVI is defined as Spartina 10-49%
5. The remainder of the shadow is defined as Mudflat
6. Classified data is added back to the polygon spatial data frame in R and written out as an ESRI Shapefile.
7. The final shapefile includes a classification of each polygon. Data can be visualized at this point and converted into a raster dataset to be stored more efficiently and analyzed.

### 8.4 Accuracy Assessment and Variable Importance (R Statistical Software)

The accuracy assessment can be done within R by computing a confusion matrix with testing data. Testing data can be partitioned from a robust training dataset. This randomly partitioned data is then reserved for testing the accuracy of the dataset. The advantages of this approach is ease of comparing multiple algorithm, variable combinations, or classification schemes (Figure 42). This approach includes the derivation of confidence intervals for overall accuracy, though the user must calculate their own Producer's and User's accuracy for further understanding of class specific accuracy. This approach was used when assessing the accuracy of FIIS and ASIS classifications. This script comes at the end of the script in Figure 41 and requires no additional input.

```r
#Create an index of training data based on the Class variable
trainIndex <- createDataPartition(sdata$Class, p=.60, list=FALSE,times=1)
#Divide the training data into training and testing data 60% as Training and 40% as Testing data
Train <- sdata@data[trainIndex,]
Test<- sdata@data[-trainIndex,]
#Next train the algorithms with the Training data
rfFit2 <- train(Class ~., data=Train, method="rf",trControl = fitcontrol, verbose=TRUE, tuneGrid=rfGrid)
#After training is conducted classify the testing data
Test$pred <- predict(rfFit2,newdata=Test)
#Calculate confusion matrix for the testing data
confusionMatrix(Test$pred,Test$Class)
#The shadow category can be removed from the testing data at least a single shadow most be within the testing data before classifying
#remove shadow
test2nosh <- test[-which(test$Class == 'Shadow'),]
#Remove shadow category from both prediction and class categories
test2nosh$Class <- factor(test2nosh$Class)
test2nosh$pred <- factor(test2nosh$pred)
```

**Figure 42.** Code block demonstrating accuracy assessment with the Caret Package in R.

After a model has been trained and the accuracy assessed it can be informative to determine the variable importance. When training models within the caret package the variable importance function can be used to assess a Random Forest mode. (Figure 43).

```r
#Assess the variable importance. Will return the 20 most importance variables.
varImp(rfFit)
```

**Figure 43.** Variable importance command in R for the classification script.

### 8.5 Merge Data and Export (R Statistical Software)

The classification can be exported from R with the writeOGR command (Figure 41, line 34).

If vector data is desired could be dissolved by class to create larger easier to manage objects.

Dissolve is found in Arctoolbox->Data Management -> Generalization -> Dissolve (Figure 44).

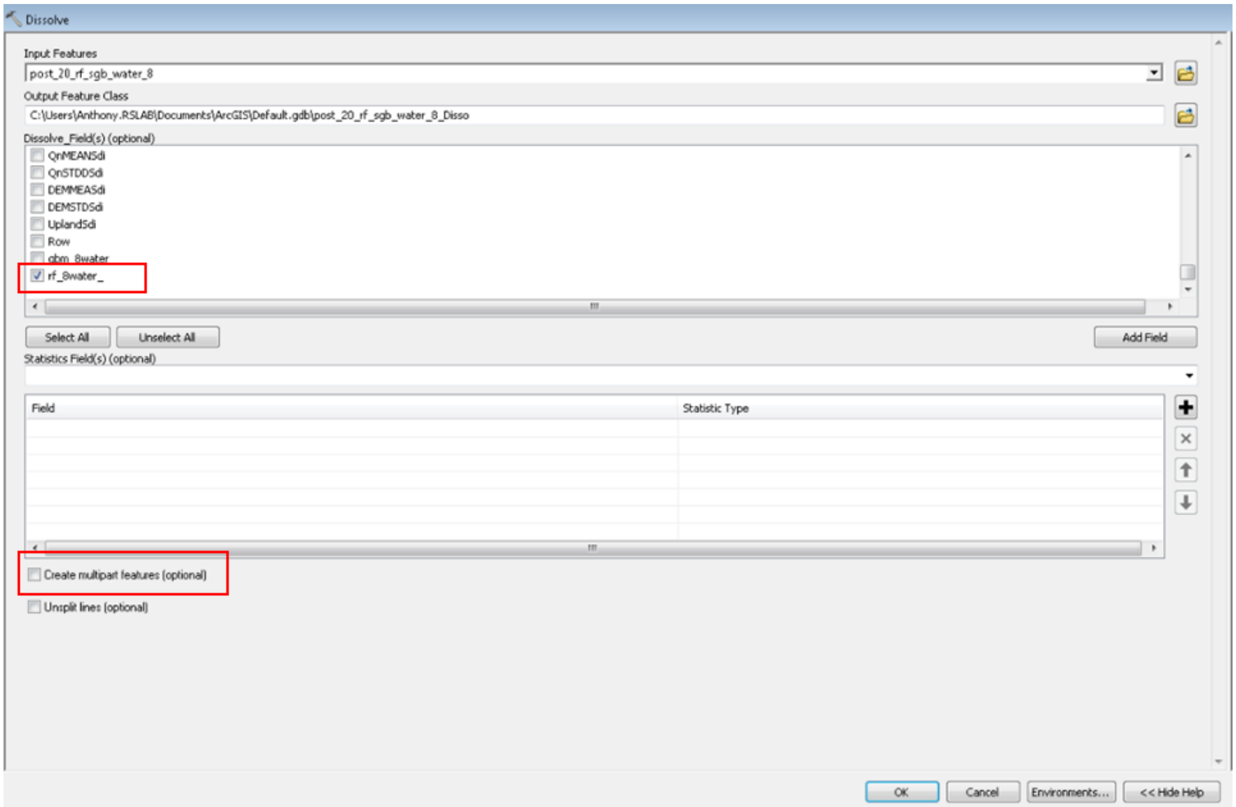



**Figure 44.** Showing the dissolve tool.

The resulting data can be converted to raster data within ArcGIS using the Feature to Raster tool found in Arctoolbox -> Conversion -> To Raster -> Feature to Raster (Figure 45).

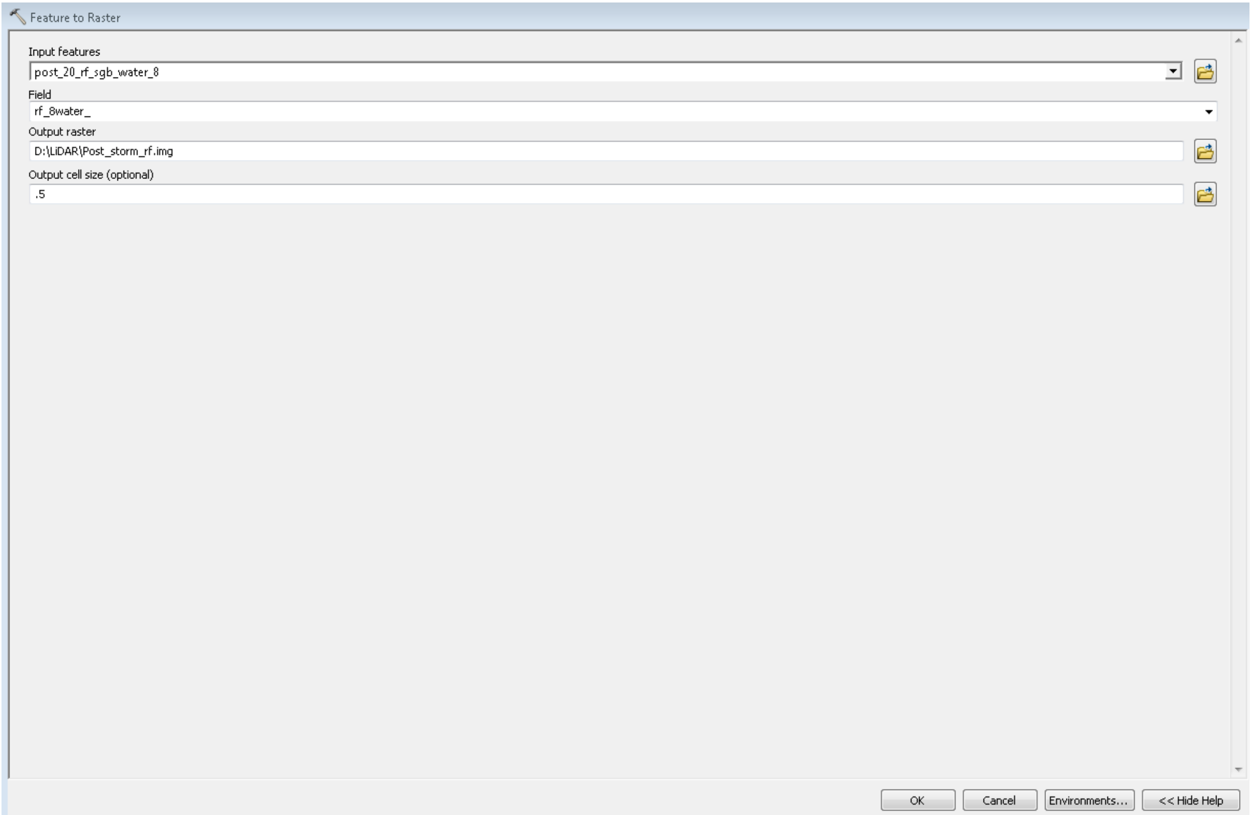



**Figure 45.** Showing the Feature to Raster tool in ArcGIS.

# 9. Accuracy Assessment (ERDAS Imagine)

Accuracy assessment is a necessary component of any remote sensing mapping project. This project utilized field knowledge and imagery to determine what class an equalized random selection of objects are (Figure 46). The accuracy assessment was conducted in ERDAS imagine.

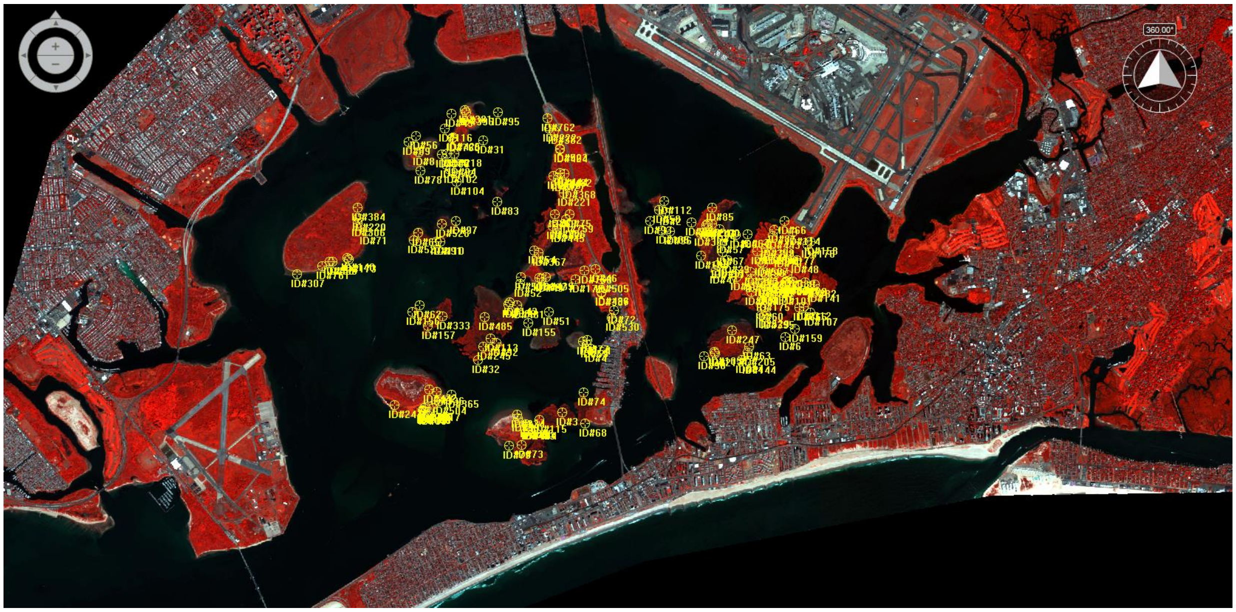


**Figure 46.** A selection of the 765 points generated to assess the Jamaica Bay classification accuracy.

**Step 1:** Ensure the data is thematic and of limited bit depth. This can be done in ERDAS Imagine with the subset tool found in the Raster -> Subset & Chip -> Create Subset Image. In red box, you can see we are saving the image as 4 bit depth and thematic (Figure 47). 4-bit depth can save up to 16 classes.

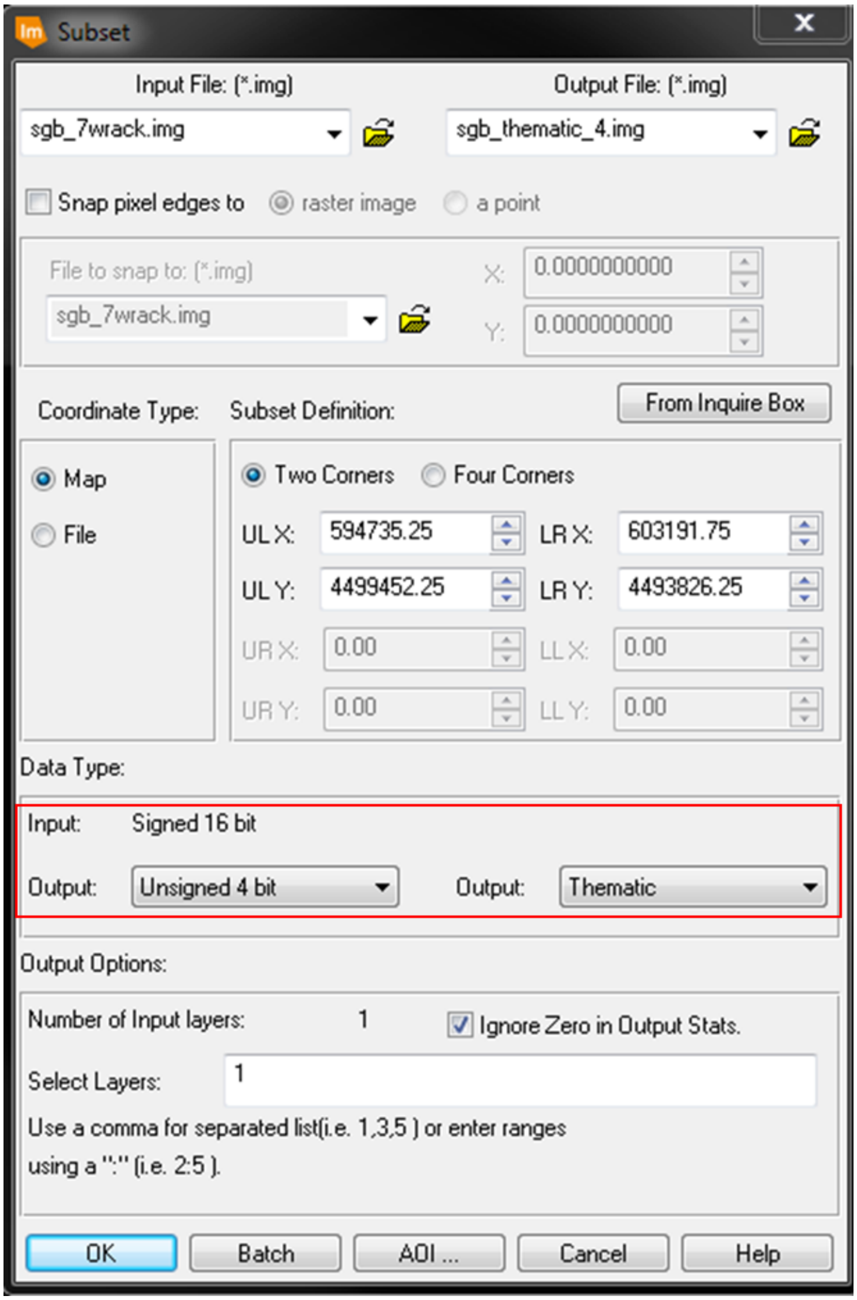



**Figure 47.** Subset an image to easily save the image as thematic with appropriate bit depth.

**Step 2:** Now that the data has been saved as thematic layer it can be used with the accuracy assessment tool. Open the accuracy assessment window as shown below. The accuracy assessment can be started from Raster -> Supervised -> Accuracy Assessment (Figure 48).

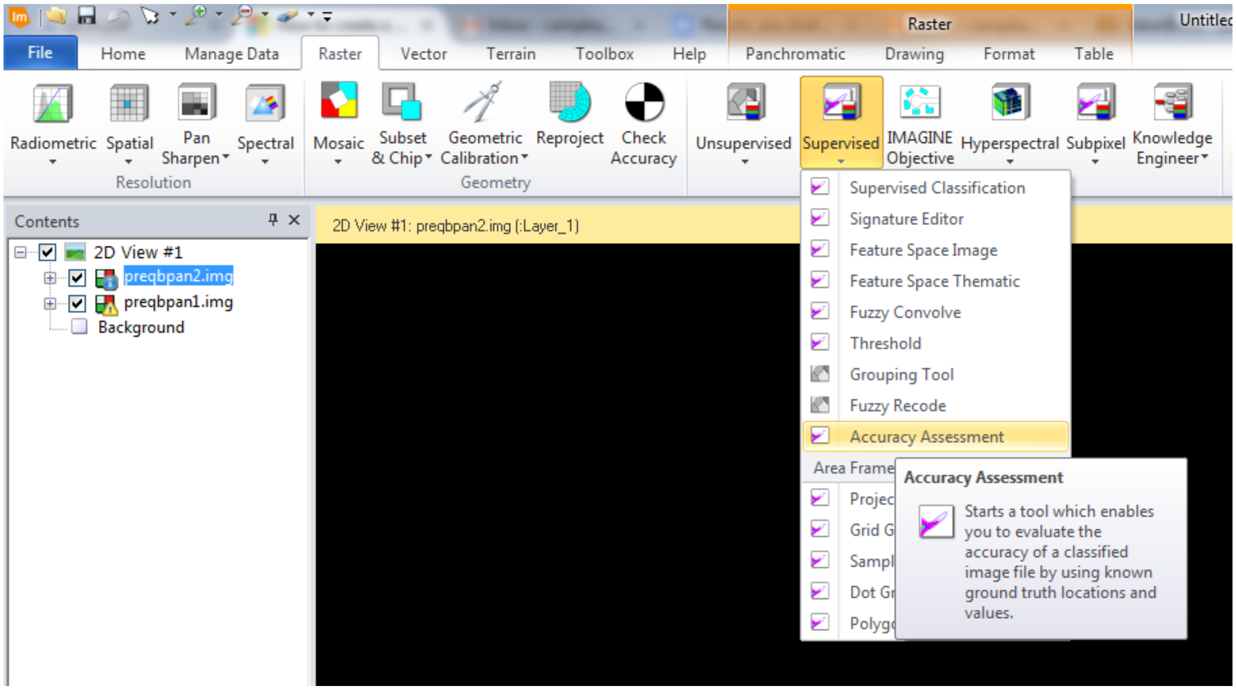



**Figure 48.** The accuracy assessment location can be seen above.

**Step 3:** Open the classification being assessed (Figure 49).

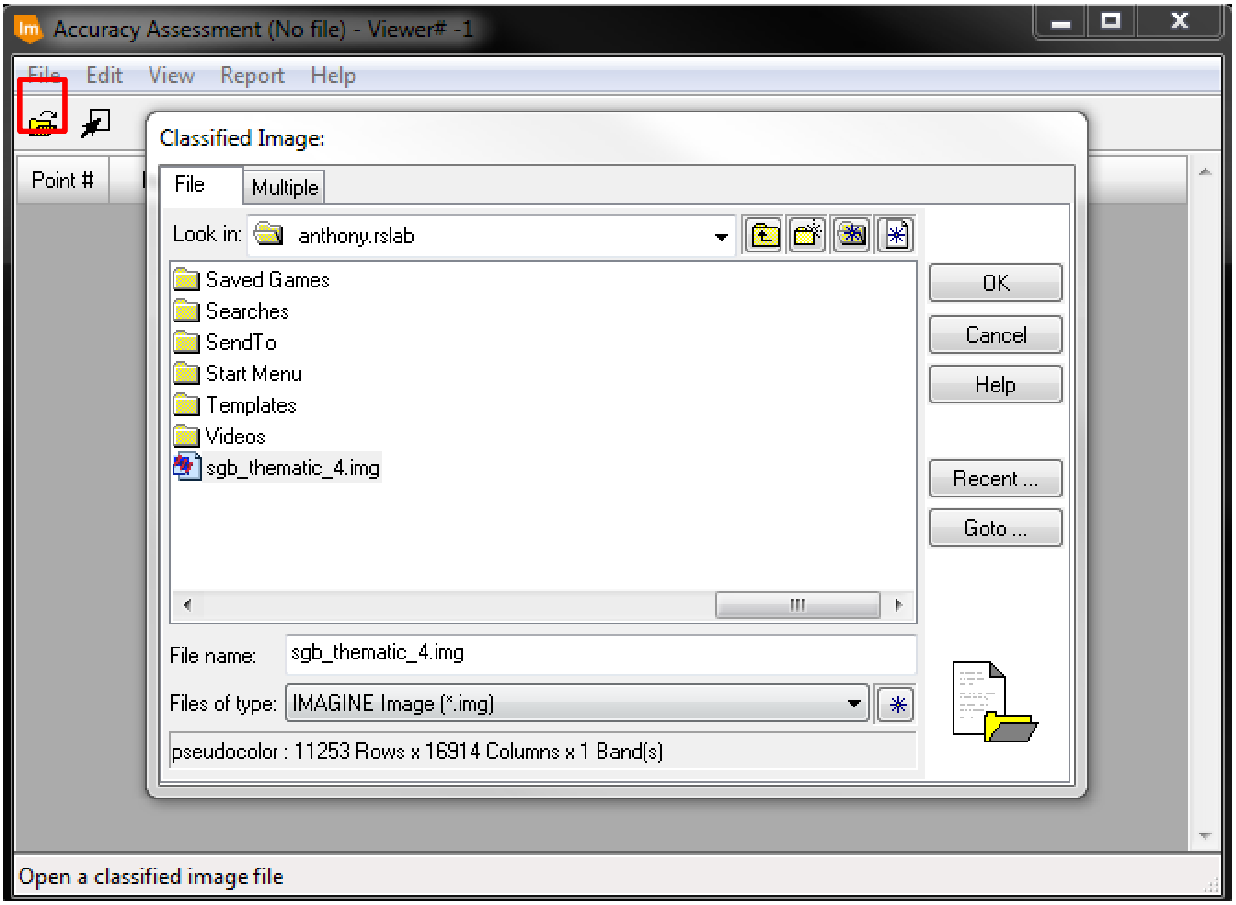



**Figure 49.** Demonstrates the opening dialog for ERDAS Accuracy Assessment.

**Step 4:** Create the equalized random points. Found in the Edit -> Create add/Random Points … Set the search count to 100,000 (Figure 50). Number of points are determined by Equation 1, and in this study was 765. Select equalized random as the distribution parameter.

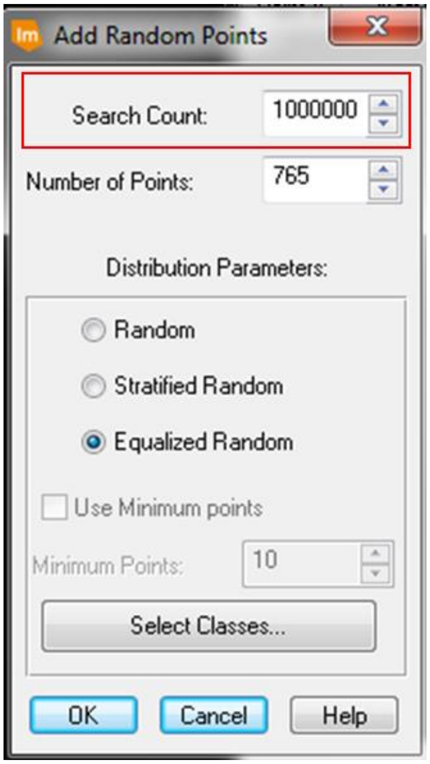
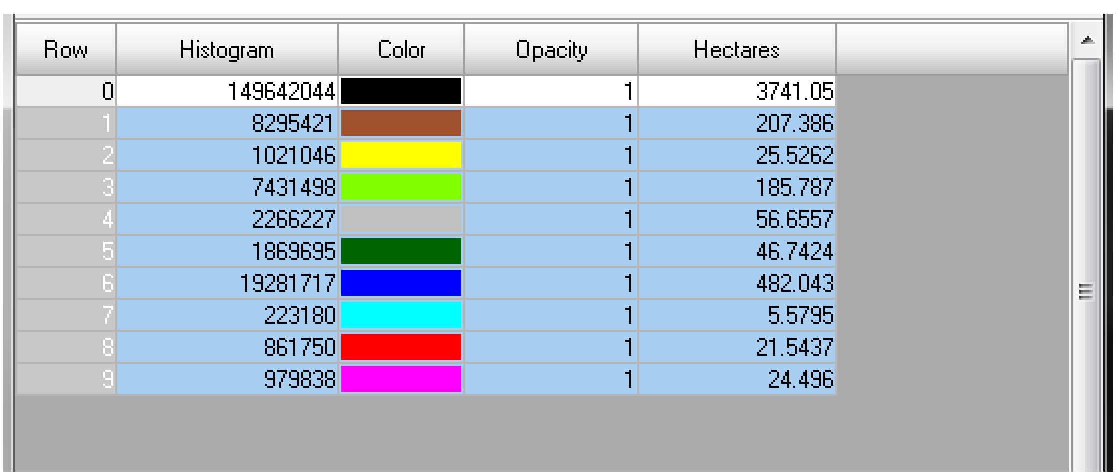



**Figure 50.** Add points dialog in the Accuracy Assessment tool for Erdas Imagine.

**Step 5:** Click select classes' button (Figure 50). Click on all classes of interest excluding background pixels (Figure 51). Once the appropriate classes have been selected click OK.



| Row | Histogram | Color | Opacity | Hectares |
|---|---|---|---|---|
| 0 | 149642044 | | 1 | 3741.05 |
| 1 | 8295421 | | 1 | 207.386 |
| 2 | 1021046 | | 1 | 25.5262 |
| 3 | 7431498 | | 1 | 185.787 |
| 4 | 2266227 | | 1 | 56.6557 |
| 5 | 1869695 | | 1 | 46.7424 |
| 6 | 19281717 | | 1 | 482.043 |
| 7 | 223180 | | 1 | 5.5795 |
| 8 | 861750 | | 1 | 21.5437 |
| 9 | 979838 | | 1 | 24.496 |

**Figure 51.** Select classes' window.

**Step 6:** Click view-> Select Viewer and click in the viewer of interest (Figure 52). Show all or select a few points and show only those. Imagery from multiple dates and the image objects should be in the viewer navigate to the point within the viewer and assess the class of that particular image object in relation to the imagery. Fill in class values in the reference category.

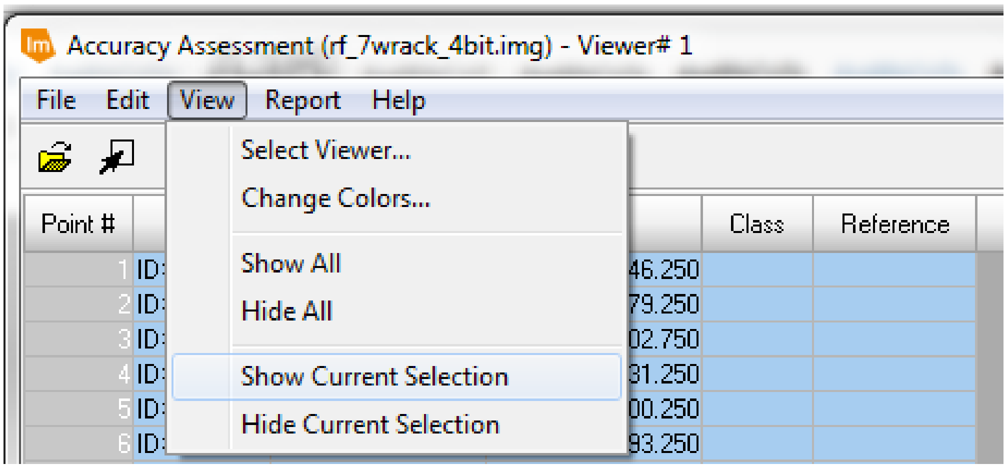



**Figure 52.** Displaying the view options within the Accuracy Assessment.

**Step 7:** Once all points have been given a class value, it is necessary to run a report. Report -> Accuracy Report. This will create a confusion matrix including Kappa, overall accuracy, user's accuracy and producer's accuracy.

# 10. Metadata

Remote sensing is an evolving field with new satellites being launched and methodologies tested frequently. This protocol utilized a variety of data both as ancillary datasets and base data used for the segmentation. An important component of understanding and continuing this protocol is the completion of detailed metadata with the classifications. The combination of metadata and this document facilitate the greater use and longevity of the classifications created with this approach. These supporting data allow users and those who undertake the classification of these sites in the future to understand and repeat the methodology.

August 2018

National Park Service
U.S. Department of the Interior

**Natural Resource Stewardship and Science**
1201 Oakridge Drive, Suite 150
Fort Collins, CO 80525